1. AWS Solutions Architect Associate - SAA-C02

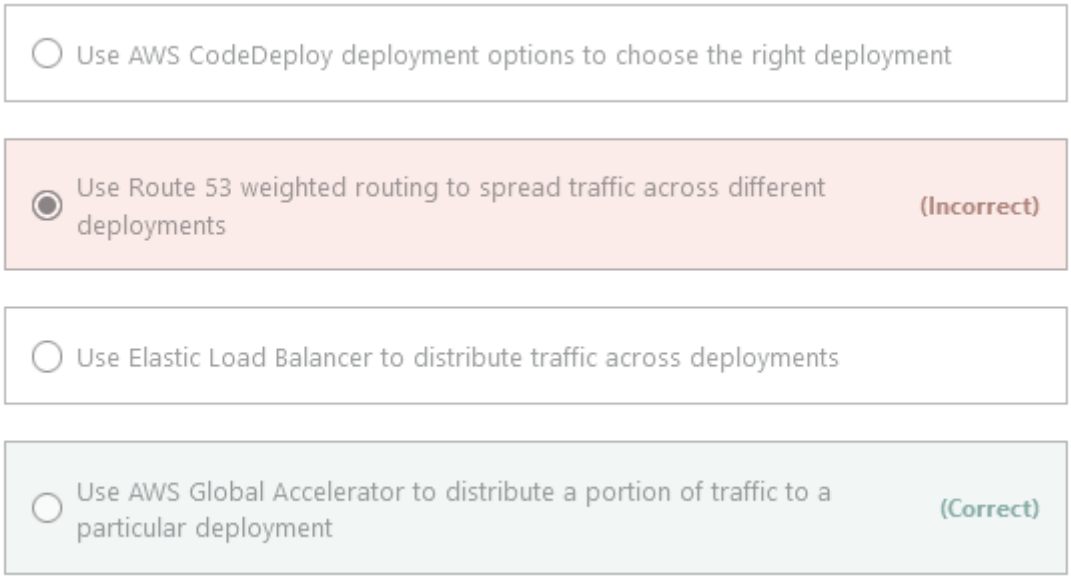
▶ Table of Contents

1.1. SAA-C02 Stephane Maarek Practice Exam

1.1.1

A retail company wants to rollout and test a blue-green deployment for its global application in the next 48 hours. Most of the customers use mobile phones which are prone to DNS caching. The company has only two days left for the annual Thanksgiving sale to commence.

As a Solutions Architect, which of the following options would you recommend to test the deployment on as many users as possible in the given time frame?



▶ Explanation

Correct option:

Blue/green deployment is a technique for releasing applications by shifting traffic between two identical environments running different versions of the application: "Blue" is the currently running version and "green" the new version. This type of deployment allows you to test features in the green environment without impacting the currently running version of your application. When you're satisfied that the green version is working properly, you can gradually reroute the traffic from the old blue environment to the new green environment. Blue/green deployments can mitigate common risks associated with deploying software, such as downtime and rollback capability.

Use AWS Global Accelerator to distribute a portion of traffic to a particular deployment - AWS Global Accelerator is a network layer service that directs traffic to optimal endpoints over the AWS global network, this improves the availability and performance of your internet applications. It provides two static anycast IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, such as your Application Load Balancers, Network Load Balancers, Elastic IP addresses or Amazon EC2 instances, in a single or in multiple AWS regions.

AWS Global Accelerator uses endpoint weights to determine the proportion of traffic that is directed to endpoints in an endpoint group, and traffic dials to control the percentage of traffic that is directed to an endpoint group (an AWS region where your application is deployed).

While relying on the DNS service is a great option for blue/green deployments, it may not fit use-cases that require a fast and controlled transition of the traffic. Some client devices and internet resolvers cache DNS answers for long periods; this DNS feature improves the efficiency of the DNS service as it reduces the DNS traffic across the Internet, and serves as a resiliency technique by preventing authoritative name-server overloads. The downside of this in blue/green deployments is that you don't know how long it will take before all of your users receive updated IP addresses when you update a record, change your routing preference or when there is an application failure.

With AWS Global Accelerator, you can shift traffic gradually or all at once between the blue and the green environment and vice-versa without being subject to DNS caching on client devices and internet resolvers, traffic dials and endpoint weights changes are effective within seconds.

Incorrect options:

Use Route 53 weighted routing to spread traffic across different deployments - 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. As discussed earlier, DNS caching is a negative behavior for this use case and hence Route 53 is not a good option.

Use Elastic Load Balancer to distribute traffic across deployments - An ELB can distribute traffic across healthy instances. You can also use the ALB weighted target groups feature for blue/green deployments as it does not rely on the DNS service. In addition you don't need to create new ALBs for the green environment. As the use-case refers to a global application, so this option cannot be used for a multi-Region solution which is needed for the given requirement.

Use AWS CodeDeploy deployment options to choose the right deployment - In CodeDeploy, a deployment is the process, and the components involved in the process, of installing content on one or more instances. This content can consist of code, web and configuration files, executables, packages, scripts, and so on. CodeDeploy deploys content that is stored in a source repository, according to the configuration rules you specify. Blue/Green deployment is one of the deployment types that CodeDeploy supports. Traffic distribution across instances, in real-time, is not a feature of CodeDeploy.

References:

https://aws.amazon.com/blogs/networking-and-content-delivery/using-aws-global-accelerator-to-achieve-blue-green-deployments

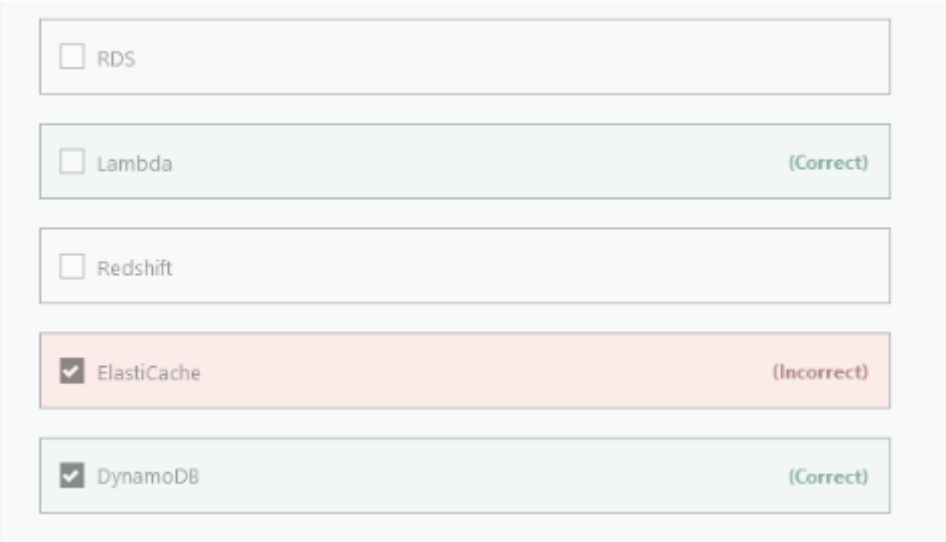
https://docs.aws.amazon.com/codedeploy/latest/userguide/deployments.html

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

1.1.2

A weather forecast agency collects key weather metrics across multiple cities in the US and sends this data in the form of key-value pairs to AWS Cloud at a one-minute frequency.

As a solutions architect, which of the following AWS services would you use to build a solution for processing and then reliably storing this data with high availability? (Select two)



▶ Explanation

Correct options:

Lambda - With AWS Lambda, you can run code without provisioning or managing servers. You pay only for the compute time that you consume—there's no charge when your code isn't running. You can run code for virtually any type of application or backend service—all with zero administration.

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 is a NoSQL database and it's best suited to store data in key-value pairs.

AWS Lambda can be combined with DynamoDB to process and capture the key-value data from the IoT sources described in the use-case. So both these options are correct.

Incorrect options:

Redshift - Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis. You cannot use Redshift to capture data in key-value pairs from the IoT sources, so this option is not correct.

ElastiCache - Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-Source compatible in-memory data stores in the cloud. Build data-intensive apps or boost the performance of your existing databases by retrieving data from high throughput and low latency in-memory data stores. Amazon ElastiCache is a popular choice for real-time use cases like Caching, Session Stores, Gaming, Geospatial Services, Real-Time Analytics, and Queuing. Elasticache is used as a caching layer in front of relational

databases. It is not a good fit to store data in key-value pairs from the IoT sources, so this option is not correct.

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. Relational databases are not a good fit to store data in key-value pairs, so this option is not correct.

References:

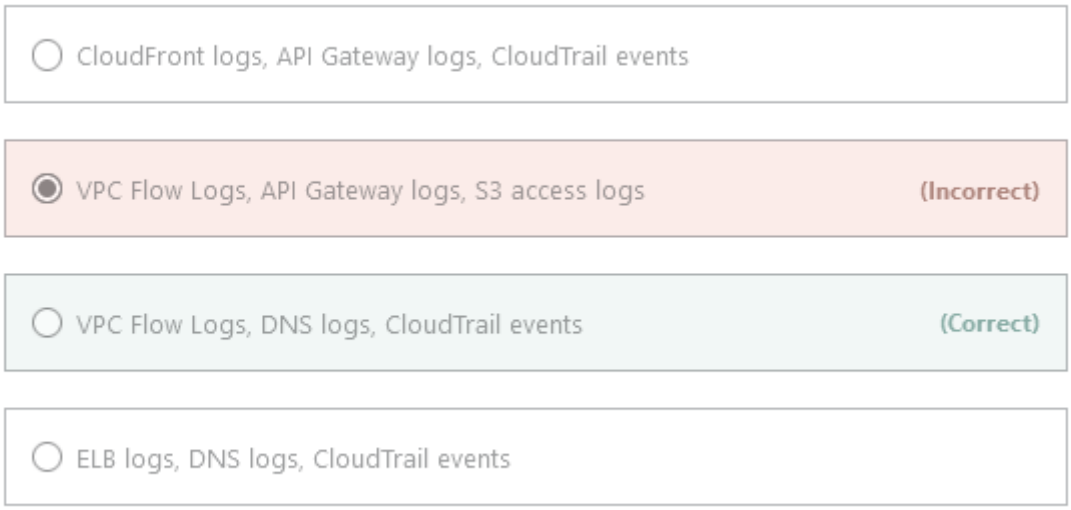
https://aws.amazon.com/dynamodb/

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

1.1.3

A retail company uses Amazon EC2 instances, API Gateway, Amazon RDS, Elastic Load Balancer and CloudFront services. To improve the security of these services, the Risk Advisory group has suggested a feasibility check for using the Amazon GuardDuty service.

Which of the following would you identify as data sources supported by GuardDuty?



▶ Explanation

Correct option:

VPC Flow Logs, DNS logs, CloudTrail events - Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3. With the cloud, the collection and aggregation of account and network activities is simplified, but it can be time-consuming for security teams to continuously analyze event log data for potential threats. With GuardDuty, you now have an intelligent and cost-effective option for continuous threat detection in AWS. The service uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats.

GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as AWS CloudTrail events, Amazon VPC Flow Logs, and DNS logs.

With a few clicks in the AWS Management Console, GuardDuty can be enabled with no software or hardware to deploy or maintain. By integrating with Amazon CloudWatch Events, GuardDuty alerts are actionable, easy to aggregate across multiple accounts, and straightforward to push into existing event management and workflow systems.

How GuardDuty works: via - https://aws.amazon.com/guardduty/

Incorrect options:

VPC Flow Logs, API Gateway logs, S3 access logs

ELB logs, DNS logs, CloudTrail events

CloudFront logs, API Gateway logs, CloudTrail events

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

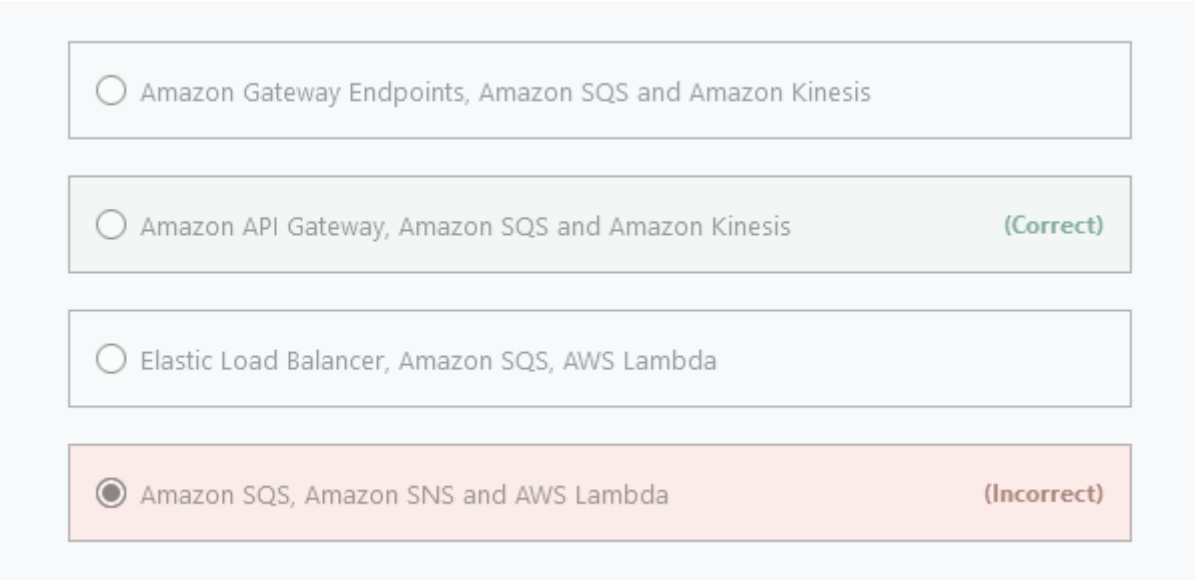
Reference:

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

1.1.4

A Big Data analytics company wants to set up an AWS cloud architecture that throttles requests in case of sudden traffic spikes. The company is looking for AWS services that can be used for buffering or throttling to handle such traffic variations.

Which of the following services can be used to support this requirement?



▶ Explanation

Correct option:

Throttling is the process of limiting the number of requests an authorized program can submit to a given operation in a given amount of time.

Amazon API Gateway, Amazon SQS and Amazon Kinesis - To prevent your API from being overwhelmed by too many requests, Amazon API Gateway throttles requests to your API using the token bucket algorithm, where a token counts for a request. Specifically, API Gateway sets a limit on a steady-state rate and a burst of request submissions against all APIs in your account. In the token bucket algorithm, the burst is the maximum bucket size.

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 offers buffer capabilities to smooth out temporary volume spikes without losing messages or increasing latency.

Amazon Kinesis - Amazon Kinesis is a fully managed, scalable service that can ingest, buffer, and process streaming data in real-time.

Incorrect options:

Amazon SQS, Amazon SNS and AWS Lambda - Amazon SQS has the ability to buffer its messages. Amazon Simple Notification Service (SNS) cannot buffer messages and is generally used with SQS to provide the buffering facility. AWS Lambda is a compute service and does not provide any buffering capability. So, this combination of services is incorrect.

Amazon Gateway Endpoints, Amazon SQS and Amazon Kinesis - A Gateway Endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service. This cannot help in throttling or buffering of requests. Amazon SQS and Kinesis can buffer incoming data. Since Gateway Endpoint is an incorrect service for throttling or buffering, this option is incorrect.

Elastic Load Balancer, Amazon SQS, AWS Lambda - Elastic Load Balancer cannot throttle requests. Amazon SQS can be used to buffer messages. AWS Lambda cannot be used for buffering. So, this combination is also incorrect.

References:

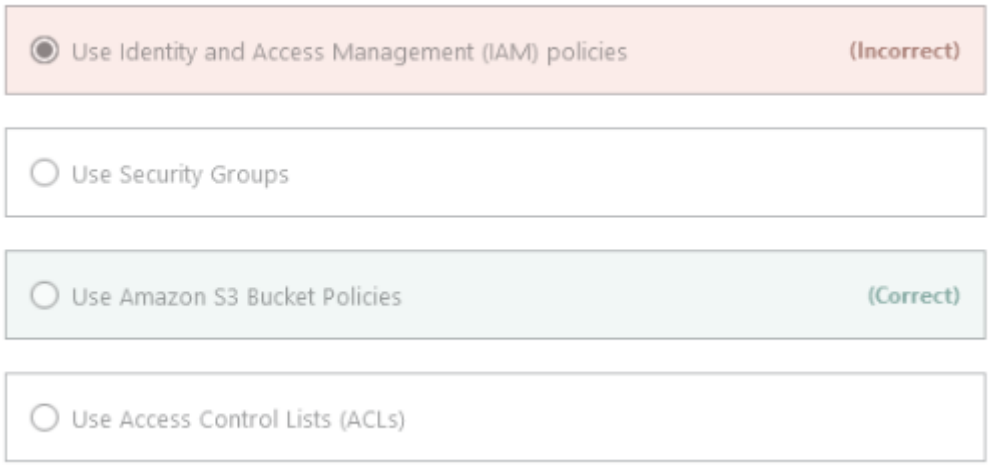
https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html

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

1.1.5

An IT company provides S3 bucket access to specific users within the same account for completing project specific work. With changing business requirements, cross-account S3 access requests are also growing every month. The company is looking for a solution that can offer user level as well as account-level access permissions for the data stored in S3 buckets.

As a Solutions Architect, which of the following would you suggest as the MOST optimized way of controlling access for this use-case?



▶ Explanation

Correct option:

Use Amazon S3 Bucket Policies

Bucket policies in Amazon S3 can be used to add or deny permissions across some or all of the objects within a single bucket. Policies can be attached to users, groups, or Amazon S3 buckets, enabling centralized management of permissions. With bucket policies, you can grant users within your AWS Account or other AWS Accounts access to your Amazon S3 resources.

You can further restrict access to specific resources based on certain conditions. For example, you can restrict access based on request time (Date Condition), whether the request was sent using SSL (Boolean Conditions), a requester's IP address (IP Address Condition), or based on the requester's client application (String Conditions). To identify these conditions, you use policy keys.

Types of access control in S3: via - https://d1.awsstatic.com/whitepapers/aws-security-whitepaper.pdf

Incorrect options:

Use Identity and Access Management (IAM) policies - AWS IAM enables organizations with many employees to create and manage multiple users under a single AWS account. IAM policies are attached to the users, enabling centralized control of permissions for users under your AWS Account to access buckets or objects. With IAM policies, you can only grant users within your own AWS account permission to access your Amazon S3 resources. So, this is not the right choice for the current requirement.

Use Access Control Lists (ACLs) - Within Amazon S3, you can use ACLs to give read or write access on buckets or objects to groups of users. With ACLs, you can only grant other AWS accounts (not specific users) access to your Amazon S3 resources. So, this is not the right choice for the current requirement.

Use Security Groups - A security group acts as a virtual firewall for EC2 instances to control incoming and outgoing traffic. S3 does not support Security Groups, this option just acts as a distractor.

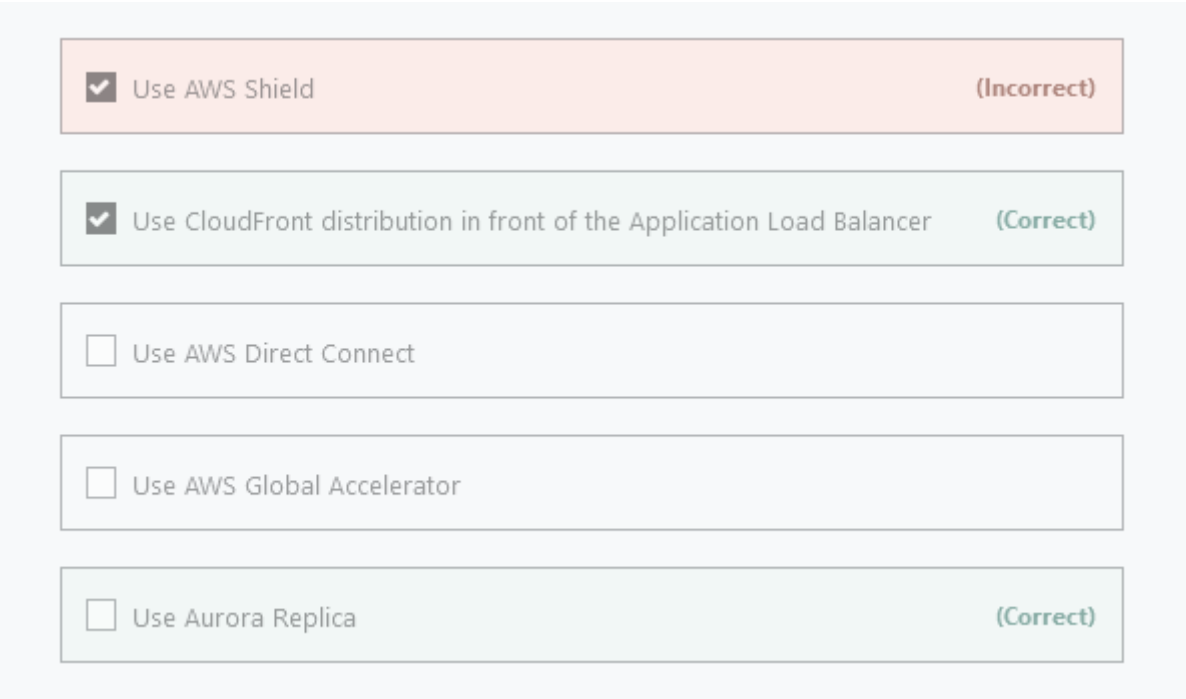
Reference:

https://d1.awsstatic.com/whitepapers/aws-security-whitepaper.pdf

1.1.6

A company manages a multi-tier social media application that runs on EC2 instances behind an Application Load Balancer. The instances run in an EC2 Auto Scaling group across multiple Availability Zones and use an Amazon Aurora database. As a solutions architect, you have been tasked to make the application more resilient to periodic spikes in request rates.

Which of the following solutions would you recommend for the given use-case? (Select two)



▶ Explanation

Correct options:

You can use Aurora replicas and CloudFront distribution to make the application more resilient to spikes in request rates.

Use Aurora Replica

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. Up to 15 Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region.

Use CloudFront distribution in front of the Application Load Balancer

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront points of presence (POPs) (edge locations) make sure that popular content can be served quickly to your viewers. CloudFront also has regional edge caches that bring more of your

content closer to your viewers, even when the content is not popular enough to stay at a POP, to help improve performance for that content.

CloudFront offers an origin failover feature to help support your data resiliency needs. CloudFront is a global service that delivers your content through a worldwide network of data centers called edge locations or points of presence (POPs). If your content is not already cached in an edge location, CloudFront retrieves it from an origin that you've identified as the source for the definitive version of the content.

Incorrect options:

Use AWS Shield* - AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency. There are two tiers of AWS Shield - Standard and Advanced. Shield cannot be used to improve application resiliency to handle spikes in traffic.

Use AWS Global Accelerator - AWS Global Accelerator is a service that improves the availability and performance of your applications with local or global users. It provides static IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, such as your Application Load Balancers, Network Load Balancers or Amazon EC2 instances. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Since CloudFront is better for improving application resiliency to handle spikes in traffic, so this option is ruled out.

Use AWS Direct Connect - 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 involve the Internet; instead, it uses dedicated, private network connections between your intranet and Amazon VPC. Direct Connect cannot be used to improve application resiliency to handle spikes in traffic.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/disaster-recovery-resiliency.html

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

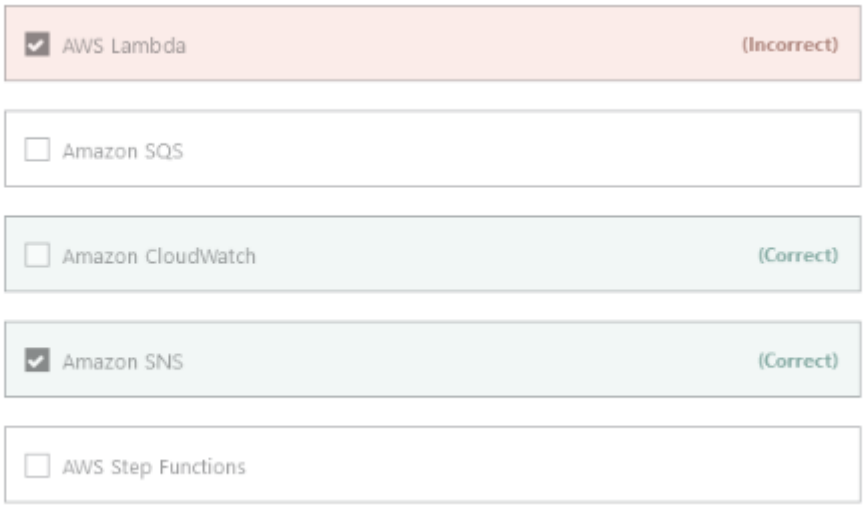
https://aws.amazon.com/global-accelerator/faqs/

https://docs.aws.amazon.com/global-accelerator/latest/dg/disaster-recovery-resiliency.html

1.1.7

A cybersecurity company uses a fleet of EC2 instances to run a proprietary application. The infrastructure maintenance group at the company wants to be notified via an email whenever the CPU utilization for any of the EC2 instances breaches a certain threshold.

Which of the following services would you use for building a solution with the LEAST amount of development effort? (Select two)



▶ Explanation

Correct options:

Amazon SNS - Amazon Simple Notification Service (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.

Amazon CloudWatch - Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. Amazon CloudWatch allows you to monitor AWS cloud resources and the applications you run on AWS.

You can use CloudWatch Alarms to send an email via SNS whenever any of the EC2 instances breaches a certain threshold. Hence both these options are correct.

Incorrect options:

AWS Lambda - With AWS Lambda, you can run code without provisioning or managing servers. You pay only for the compute time that you consume—there's no charge when your code isn't running. You can run code for virtually any type of application or backend service—all with zero administration. You cannot use AWS Lambda to monitor CPU utilization of EC2 instances or send notification emails, hence this option is incorrect.

Amazon SQS - Amazon SQS Standard offers a reliable, highly scalable hosted queue for storing messages as they travel between computers. Amazon SQS lets you easily move data between distributed application components and helps you build applications in which messages are processed independently (with message-

level ack/fail semantics), such as automated workflows. You cannot use SQS to monitor CPU utilization of EC2 instances or send notification emails, hence this option is incorrect.

AWS Step Functions - AWS Step Functions lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Using Step Functions, you can design and run workflows that stitch together services, such as AWS Lambda, AWS Fargate, and Amazon SageMaker, into feature-rich applications. You cannot use Step Functions to monitor CPU utilization of EC2 instances or send notification emails, hence this option is incorrect.

References:

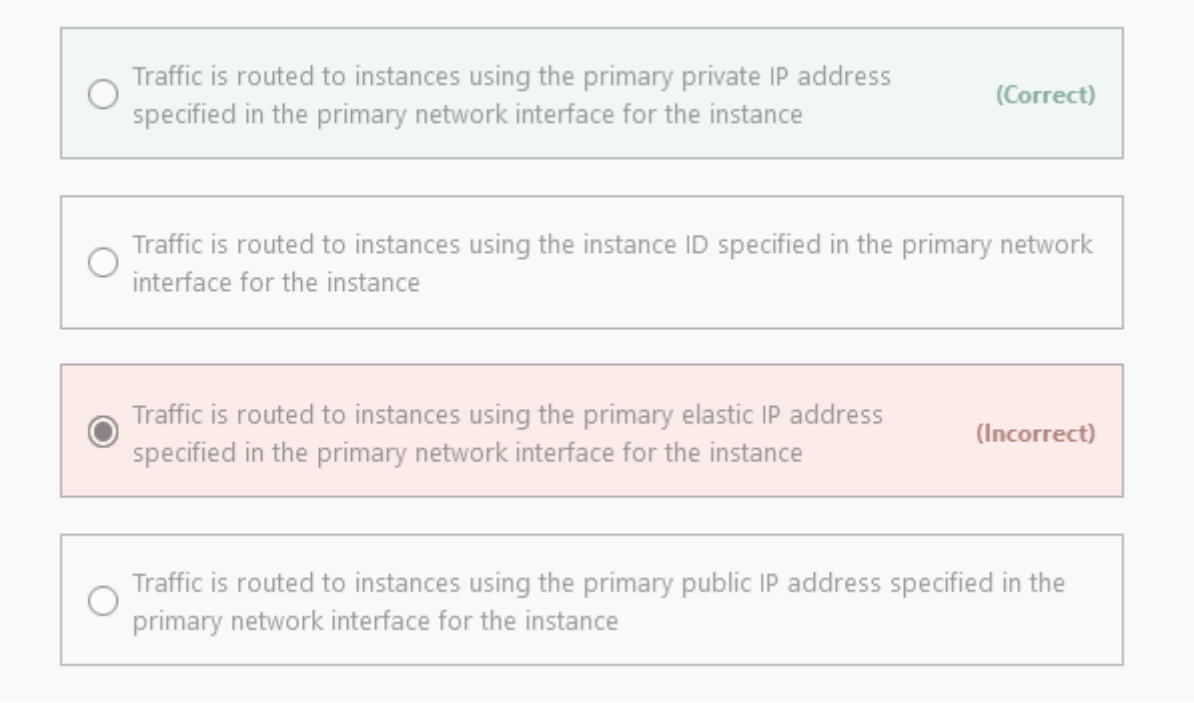
https://aws.amazon.com/cloudwatch/faqs/

https://aws.amazon.com/sns/

1.1.8

A leading online gaming company is migrating its flagship application to AWS Cloud for delivering its online games to users across the world. The company would like to use a Network Load Balancer (NLB) to handle millions of requests per second. The engineering team has provisioned multiple instances in a public subnet and specified these instance IDs as the targets for the NLB.

As a solutions architect, can you help the engineering team understand the correct routing mechanism for these target instances?



▶ Explanation

Correct option:

Traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.

Request Routing and IP Addresses -

If you specify targets using an instance ID, traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance. The load balancer rewrites the destination IP address from the data packet before forwarding it to the target instance.

If you specify targets using IP addresses, you can route traffic to an instance using any private IP address from one or more network interfaces. This enables multiple applications on an instance to use the same port. Note that each network interface can have its security group. The load balancer rewrites the destination IP address before forwarding it to the target.

Incorrect options: Traffic is routed to instances using the primary public IP address specified in the primary network interface for the instance - If you specify targets using an instance ID, traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance. So public IP address cannot be used to route the traffic to the instance.

Traffic is routed to instances using the primary elastic IP address specified in the primary network interface for the instance - If you specify targets using an instance ID, traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance. So elastic IP address cannot be used to route the traffic to the instance.

Traffic is routed to instances using the instance ID specified in the primary network interface for the instance - You cannot use instance ID to route traffic to the instance. This option is just added as a distractor.

References:

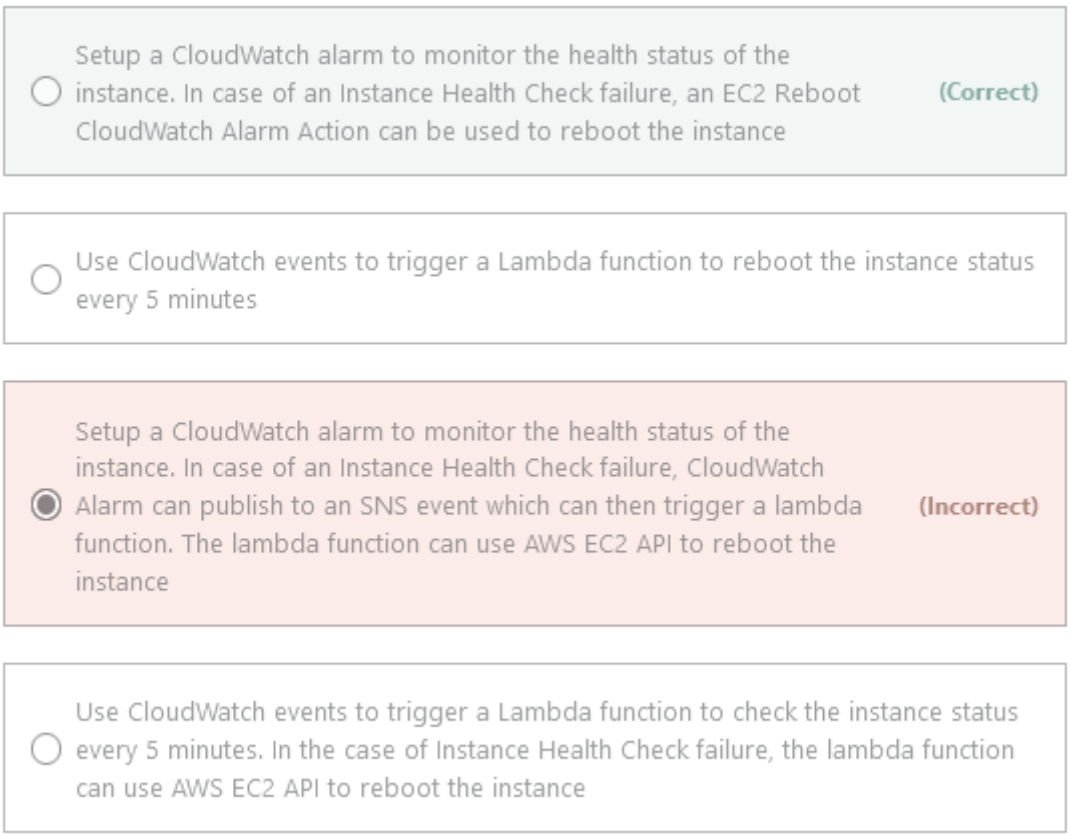
https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-target-groups.html

1.1.9

A medium-sized business has a taxi dispatch application deployed on an 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?



▶ Explanation

Correct option:

Setup a 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 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 a CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, CloudWatch Alarm can publish to an SNS event which can then trigger a lambda function. The lambda function can use AWS EC2 API to reboot the instance

Use CloudWatch events to trigger a Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the lambda function can use AWS EC2 API to reboot the instance

Use CloudWatch events to trigger a Lambda function to reboot the instance status every 5 minutes

Using CloudWatch event or CloudWatch alarm to trigger a 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 lambda function are incorrect.

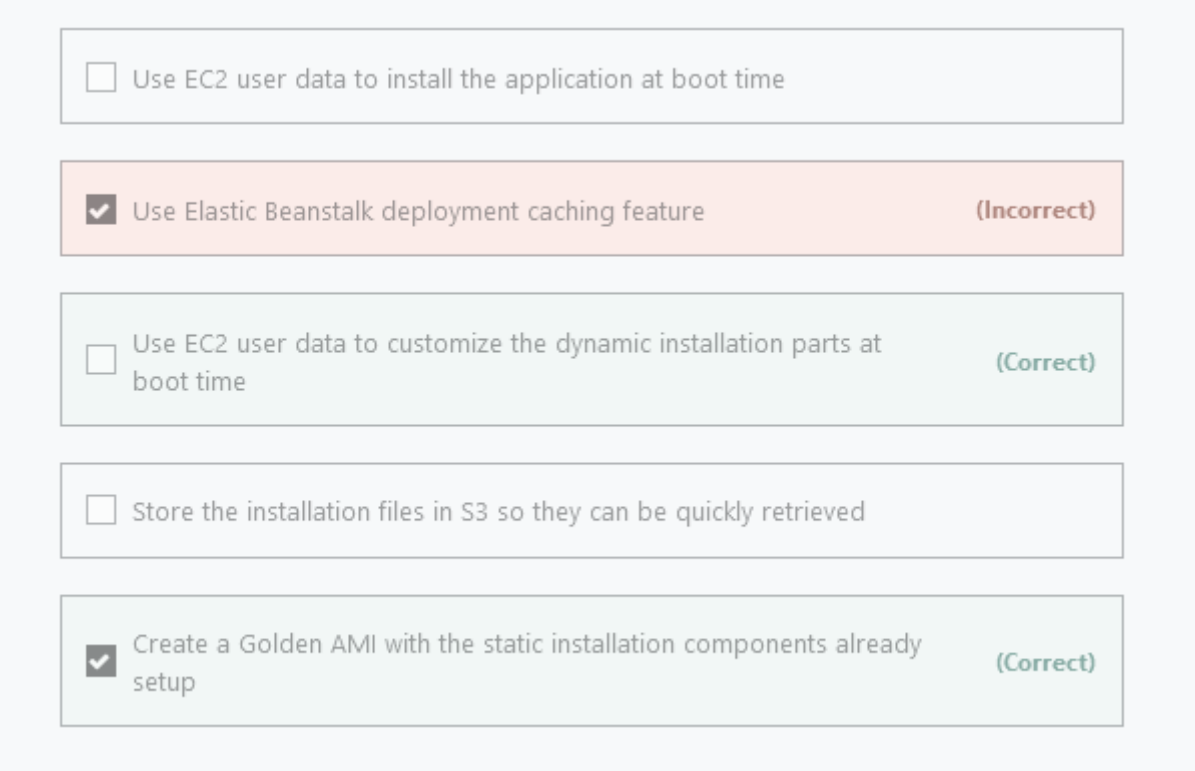
Reference:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/UsingAlarmActions.html

1.1.10

Your company is deploying a website running on Elastic Beanstalk. The website takes over 45 minutes for the installation and contains both static as well as dynamic files that must be generated during the installation process.

As a Solutions Architect, you would like to bring the time to create a new Instance in your Elastic Beanstalk deployment to be less than 2 minutes. What do you recommend? (Select two)



▶ Explanation

Correct option:

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

When you create an AWS Elastic Beanstalk environment, you can specify an Amazon Machine Image (AMI) to use instead of the standard Elastic Beanstalk AMI included in your platform version. A custom AMI can improve provisioning times when instances are launched in your environment if you need to install a lot of software that isn't included in the standard AMIs.

Create a Golden AMI with the static installation components already setup - A Golden AMI is an AMI that you standardize through configuration, consistent security patching, and hardening. It also contains agents you

approve for logging, security, performance monitoring, etc. For the given use-case, you can have the static installation components already setup via the golden AMI.

Use EC2 user data to customize the dynamic installation parts at boot time - EC2 instance user data is the data that you specified in the form of a configuration script while launching your instance. You can use EC2 user data to customize the dynamic installation parts at boot time, rather than installing the application itself at boot time.

Incorrect options:

Store the installation files in S3 so they can be quickly retrieved - Amazon S3 bucket can be used as a storage location for your source code, logs, and other artifacts that are created when you use Elastic Beanstalk. It cannot be used to run or generate dynamic files since S3 is not an environment but a storage service.

Use EC2 user data to install the application at boot time - User data of an instance can be used to perform common automated configuration tasks or run scripts after the instance starts. User data, cannot, however, be used to install the application.

Use Elastic Beanstalk deployment caching feature - Elastic Beanstalk deployment caching is a made-up option. It is just added as a distractor.

References:

https://aws.amazon.com/elasticbeanstalk/

https://aws.amazon.com/blogs/awsmarketplace/announcing-the-golden-ami-pipeline/

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/AWSHowTo.S3.html

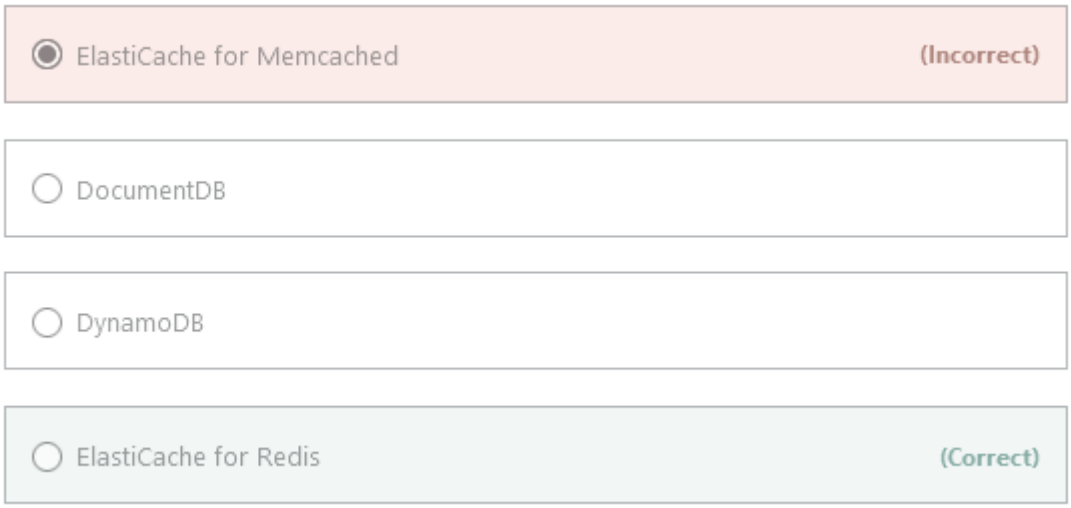
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.customenv.html

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

1.1.11

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 an in-memory database that is highly available as well as HIPAA compliant.

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



▶ Explanation

Correct option:

ElastiCache for Redis

ElastiCache Overview: 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 Redis is also HIPAA Eligible Service. Therefore, this is the correct option.

ElastiCache for Redis Overview: via - https://aws.amazon.com/elasticache/redis/

Exam Alert:

Please review this comparison sheet for Redis vs Memcached features: via - https://aws.amazon.com/elasticache/redis-vs-memcached/

Incorrect Options:

ElastiCache for Memcached - 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. Elasticache for Memcached is not HIPAA eligible, so this option is incorrect.

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. DynamoDB is not an in-memory database, so this option is incorrect.

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. DocumentDB is not an in-memory database, so this option is incorrect.

References:

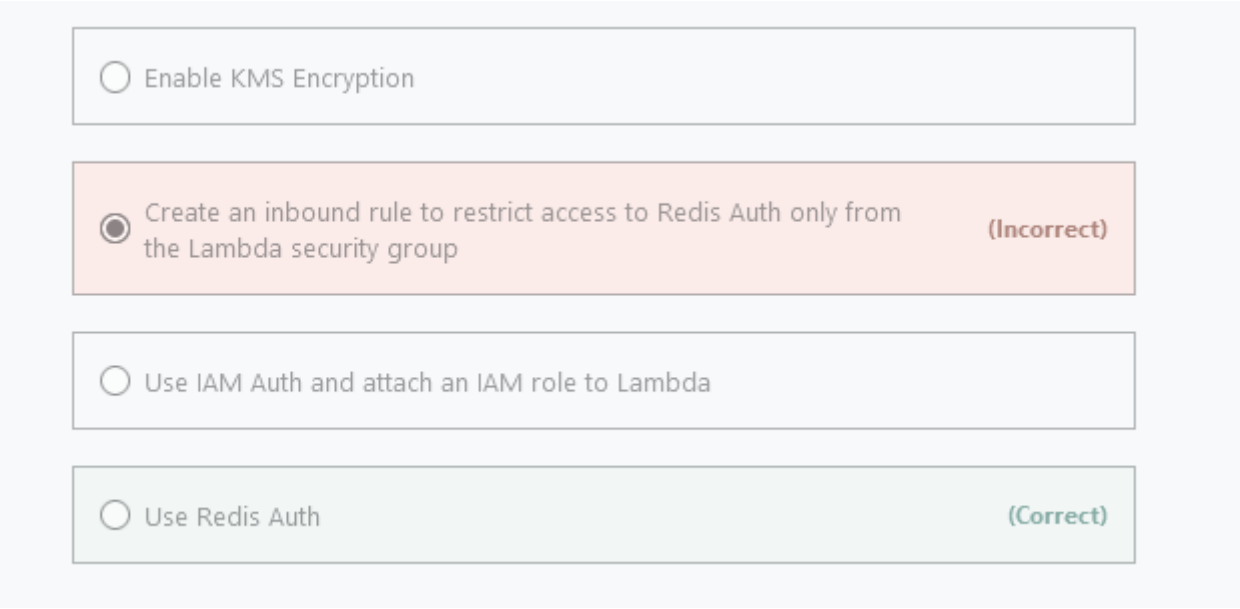
https://aws.amazon.com/about-aws/whats-new/2017/11/amazon-elasticache-for-redis-is-now-hipaa-eligible-to-help-you-power-secure-healthcare-applications-with-sub-millisecond-latency/

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

1.1.12

Your company runs a website for evaluating coding skills. As a Solutions Architect, you've designed the architecture of the website to follow a serverless pattern on the AWS Cloud using API Gateway and AWS Lambda. The backend is using an RDS PostgreSQL database. Caching is implemented using a Redis ElastiCache cluster. You would like to increase the security of your authentication to Redis from the Lambda function, leveraging a username and password combination.

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



▶ Explanation

Correct option:

Use Redis Auth - 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. IAM Auth is not supported by ElastiCache.

Redis authentication tokens enable Redis to require a token (password) before allowing clients to execute commands, thereby improving data security.

Incorrect options:

Use IAM Auth and attach an IAM role to Lambda - As discussed above, IAM Auth is not supported by ElastiCache.

Enable KMS Encryption - AWS Key Management Service (KMS) makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. AWS

KMS is a secure and resilient service that uses hardware security modules that have been validated under FIPS 140-2. KMS does not support username and password for enabling encryption.

Create an inbound rule to restrict access to Redis Auth only from the Lambda security group - A security group acts as a virtual firewall that controls the traffic for one or more instances. You can add rules to each security group that allows 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.

References:

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

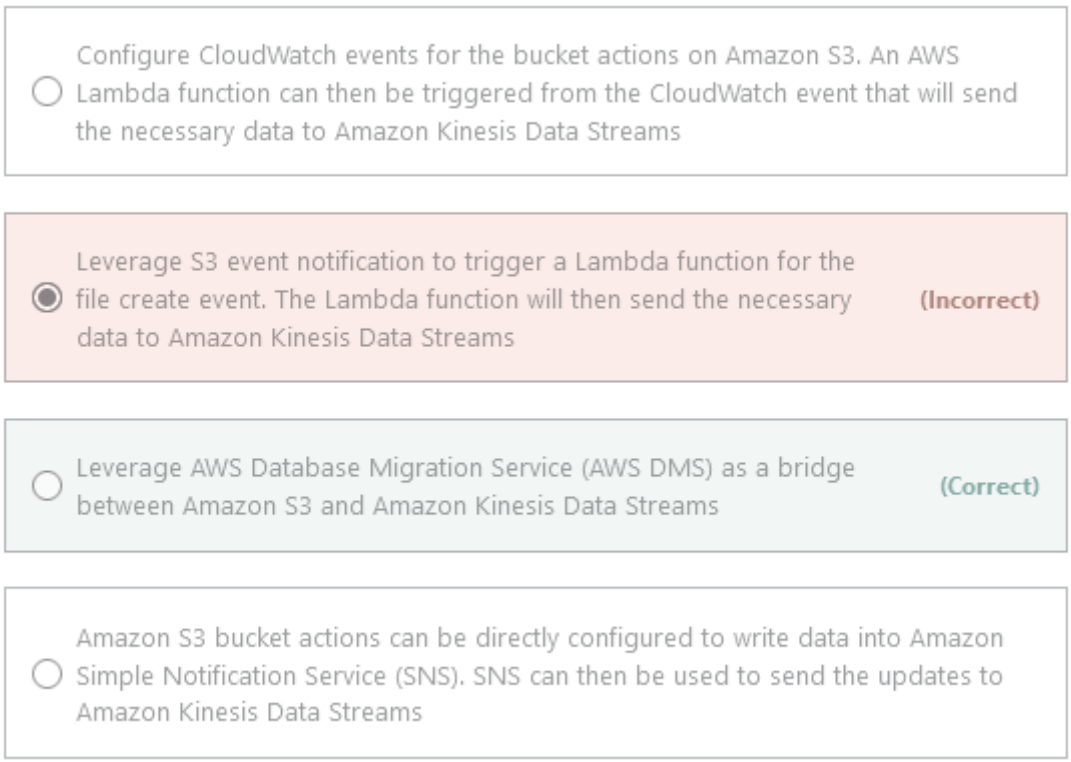
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/IAM.Overview.html

https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html

1.1.13

A Big Data analytics company writes data and log files in Amazon S3 buckets. The company now wants to stream the existing data files as well as any ongoing file updates from Amazon S3 to Amazon Kinesis Data Streams.

As a Solutions Architect, which of the following would you suggest as the fastest possible way of building a solution for this requirement?



▶ Explanation

Correct option:

Leverage AWS Database Migration Service (AWS DMS) as a bridge between Amazon S3 and Amazon Kinesis Data Streams - You can achieve this by using AWS Database Migration Service (AWS DMS). AWS DMS enables you to seamlessly migrate data from supported sources to relational databases, data warehouses, streaming platforms, and other data stores in AWS cloud.

The given requirement needs the functionality to be implemented in the least possible time. You can use AWS DMS for such data-processing requirements. AWS DMS lets you expand the existing application to stream data from Amazon S3 into Amazon Kinesis Data Streams for real-time analytics without writing and maintaining new code. AWS DMS supports specifying Amazon S3 as the source and streaming services like Kinesis and Amazon Managed Streaming of Kafka (Amazon MSK) as the target. AWS DMS allows migration of full and change data capture (CDC) files to these services. AWS DMS performs this task out of box without any complex configuration or code development. You can also configure an AWS DMS replication instance to scale up or down depending on the workload.

AWS DMS supports Amazon S3 as the source and Kinesis as the target, so data stored in an S3 bucket is streamed to Kinesis. Several consumers, such as AWS Lambda, Amazon Kinesis Data Firehose, Amazon Kinesis

Data Analytics, and the Kinesis Consumer Library (KCL), can consume the data concurrently to perform real-time analytics on the dataset. Each AWS service in this architecture can scale independently as needed.

Architecture of the proposed solution: via - https://aws.amazon.com/blogs/big-data/streaming-data-from-amazon-s3-to-amazon-kinesis-data-streams-using-aws-dms/

Incorrect options:

Configure CloudWatch events for the bucket actions on Amazon S3. An AWS Lambda function can then be triggered from the CloudWatch event that will send the necessary data to Amazon Kinesis Data Streams - You will need to enable a Cloudtrail trail to use object-level actions as a trigger for CloudWatch events. Also, using Lambda functions would require significant custom development to write the data into Kinesis Data Streams, so this option is not the right fit.

Leverage S3 event notification to trigger a Lambda function for the file create event. The Lambda function will then send the necessary data to Amazon Kinesis Data Streams - Using Lambda functions would require significant custom development to write the data into Kinesis Data Streams, so this option is not the right fit.

Amazon S3 bucket actions can be directly configured to write data into Amazon Simple Notification Service (SNS). SNS can then be used to send the updates to Amazon Kinesis Data Streams - S3 cannot directly write data into SNS, although it can certainly use S3 event notifications to send an event to SNS. Also, SNS cannot directly send messages to Kinesis Data Streams. So this option is incorrect.

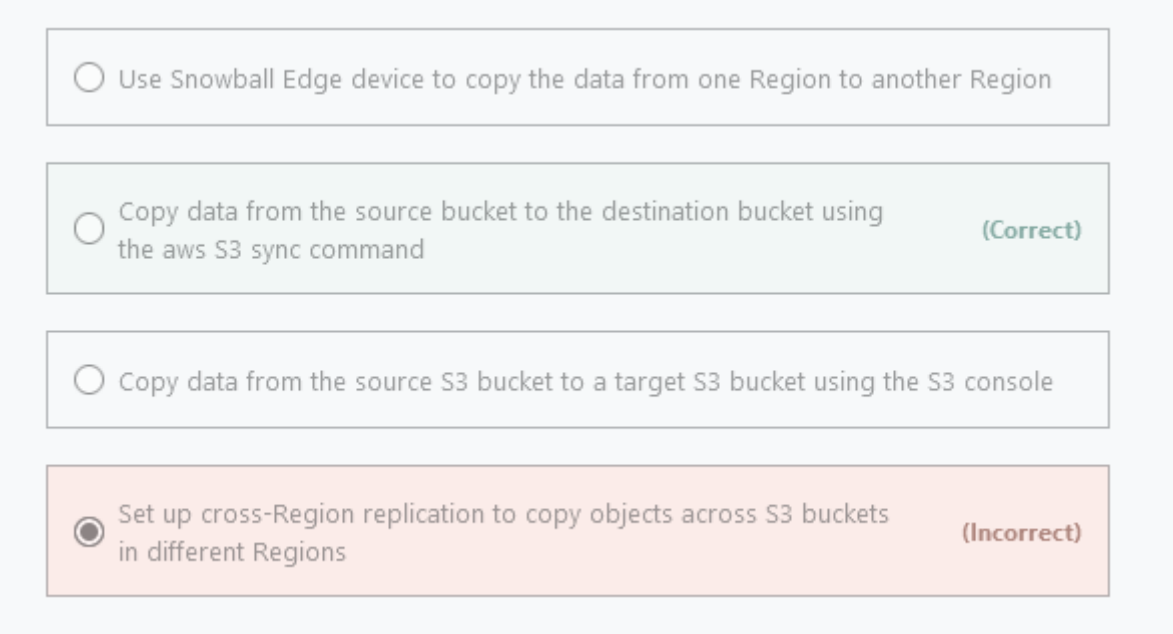
Reference:

https://aws.amazon.com/blogs/big-data/streaming-data-from-amazon-s3-to-amazon-kinesis-data-streams-using-aws-dms/

1.1.14

An e-commerce company has copied 1 PB of data from its on-premises data center to an Amazon S3 bucket in the us-west-1 Region using an AWS Direct Connect link. The company now wants to copy the data to another S3 bucket in the us-east-1 Region. The on-premises data center does not allow the use of AWS Snowball.

As a Solutions Architect, which of the following would you recommend to accomplish this?



▶ Explanation

Correct options:

Copy data from the source bucket to the destination bucket using the aws S3 sync command

The aws S3 sync command uses the CopyObject APIs to copy objects between S3 buckets. The sync command lists the source and target buckets to identify objects that are in the source bucket but that aren't in the target bucket. The command also identifies objects in the source bucket that have different LastModified dates than the objects that are in the target bucket. The sync command on a versioned bucket copies only the current version of the object—previous versions aren't copied. By default, this preserves object metadata, but the access control lists (ACLs) are set to FULL_CONTROL for your AWS account, which removes any additional ACLs. If the operation fails, you can run the sync command again without duplicating previously copied objects.

You can use the command like so:

aws s3 sync s3://DOC-EXAMPLE-BUCKET-SOURCE s3://DOC-EXAMPLE-BUCKET-TARGET

Incorrect options:

Use Snowball Edge device to copy the data from one Region to another Region - As the given requirement is about copying the data from one AWS Region to another AWS Region, so Snowball Edge cannot be used here. Snowball Edge Storage Optimized is the optimal data transfer choice if you need to securely and quickly transfer terabytes to petabytes of data to AWS. You can use Snowball Edge Storage Optimized if you have a

large backlog of data to transfer or if you frequently collect data that needs to be transferred to AWS and your storage is in an area where high-bandwidth internet connections are not available or cost-prohibitive. Snowball Edge can operate in remote locations or harsh operating environments, such as factory floors, oil and gas rigs, mining sites, hospitals, and on moving vehicles.

Copy data from the source S3 bucket to a target S3 bucket using the S3 console - AWS S3 console cannot be used to transfer 1PB of data from one bucket to another as it's not feasible. You must use S3 sync for this requirement.

Set up cross-Region replication to copy objects across S3 buckets in different Regions - As the data already exists in the source bucket, so you cannot use cross-Region replication because, by default, replication only supports copying new Amazon S3 objects after it is enabled. Replication enables automatic, asynchronous copying of objects across Amazon S3 buckets. Buckets that are configured for object replication can be owned by the same AWS account or by different accounts. Object may be replicated to a single destination bucket or multiple destination buckets. Destination buckets can be in different AWS Regions or within the same Region as the source bucket.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/move-objects-s3-bucket/

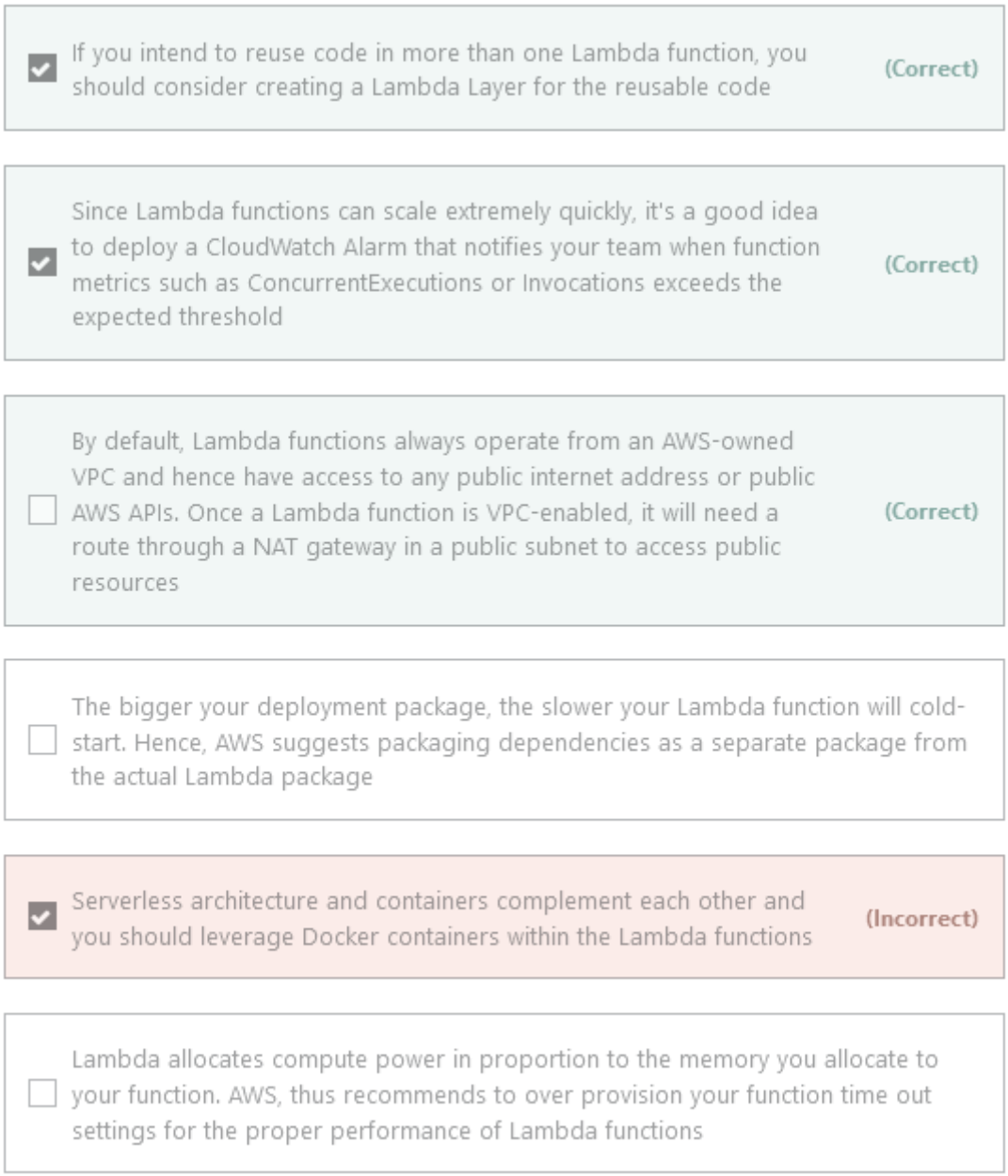
https://aws.amazon.com/snowball/faqs/

https://docs.aws.amazon.com/AmazonS3/latest/userguide/replication.html

1.1.15

The engineering team at an e-commerce company has been tasked with migrating to a serverless architecture. The team wants to focus on the key points of consideration when using Lambda as a backbone for this architecture.

As a Solutions Architect, which of the following options would you identify as correct for the given requirement? (Select three)



▶ Explanation

Correct options:

By default, Lambda functions always operate from an AWS-owned VPC and hence have access to any public internet address or public AWS APIs. Once a Lambda function is VPC-enabled, it will need a route through a NAT gateway in a public subnet to access public resources - Lambda functions always operate from an AWS-

owned VPC. By default, your function has the full ability to make network requests to any public internet address — this includes access to any of the public AWS APIs. For example, your function can interact with AWS DynamoDB APIs to PutItem or Query for records. You should only enable your functions for VPC access when you need to interact with a private resource located in a private subnet. An RDS instance is a good example.

Once your function is VPC-enabled, all network traffic from your function is subject to the routing rules of your VPC/Subnet. If your function needs to interact with a public resource, you will need a route through a NAT gateway in a public subnet.

When to VPC-Enable a Lambda Function: via - https://aws.amazon.com/blogs/architecture/best-practices-for-developing-on-aws-lambda/

Since Lambda functions can scale extremely quickly, its a good idea to deploy a CloudWatch Alarm that notifies your team when function metrics such as ConcurrentExecutions or Invocations exceeds the expected threshold - Since Lambda functions can scale extremely quickly, this means you should have controls in place to notify you when you have a spike in concurrency. A good idea is to deploy a CloudWatch Alarm that notifies your team when function metrics such as ConcurrentExecutions or Invocations exceeds your threshold. You should create an AWS Budget so you can monitor costs on a daily basis.

If you intend to reuse code in more than one Lambda function, you should consider creating a Lambda Layer for the reusable code - You can configure your Lambda function to pull in additional code and content in the form of layers. A layer is a ZIP archive that contains libraries, a custom runtime, or other dependencies. With layers, you can use libraries in your function without needing to include them in your deployment package. Layers let you keep your deployment package small, which makes development easier. A function can use up to 5 layers at a time.

You can create layers, or use layers published by AWS and other AWS customers. Layers support resource-based policies for granting layer usage permissions to specific AWS accounts, AWS Organizations, or all accounts. The total unzipped size of the function and all layers can't exceed the unzipped deployment package size limit of 250 MB.

Incorrect options:

Lambda allocates compute power in proportion to the memory you allocate to your function. AWS, thus recommends to over provision your function time out settings for the proper performance of Lambda functions - Lambda allocates compute power in proportion to the memory you allocate to your function. This means you can over-provision memory to run your functions faster and potentially reduce your costs. However, AWS recommends that you should not over provision your function time out settings. Always understand your code performance and set a function time out accordingly. Overprovisioning function timeout often results in Lambda functions running longer than expected and unexpected costs.

The bigger your deployment package, the slower your Lambda function will cold-start. Hence, AWS suggests packaging dependencies as a separate package from the actual Lambda package - This statement is incorrect and acts as a distractor. All the dependencies are also packaged into the single Lambda deployment package.

Serverless architecture and containers complement each other and you should leverage Docker containers within the Lambda functions - This statement is incorrect. AWS Lambda does not support running Docker containers.

References:

https://aws.amazon.com/blogs/architecture/best-practices-for-developing-on-aws-lambda/

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

1.1.16

An Electronic Design Automation (EDA) application produces massive volumes of data that can be divided into two categories. The 'hot data' needs to be both processed and stored quickly in a parallel and distributed fashion. The 'cold data' needs to be kept for reference with quick access for reads and updates at a low cost.

Which of the following AWS services is BEST suited to accelerate the aforementioned chip design process?

▶ Explanation

Correct option:

Amazon FSx for Lustre

Amazon FSx for Lustre makes it easy and cost-effective to launch and run the world's most popular high-performance file system. It is used for workloads such as machine learning, high-performance computing (HPC), video processing, and financial modeling. The open-source Lustre file system is designed for applications that require fast storage – where you want your storage to keep up with your compute. FSx for Lustre integrates with Amazon S3, making it easy to process data sets with the Lustre file system. When linked to an S3 bucket, an FSx for Lustre file system transparently presents S3 objects as files and allows you to write changed data back to S3.

FSx for Lustre provides the ability to both process the 'hot data' in a parallel and distributed fashion as well as easily store the 'cold data' on Amazon S3. Therefore this option is the BEST fit for the given problem statement.

Incorrect options:

Amazon FSx for Windows File Server - Amazon FSx for Windows File Server provides fully managed, highly reliable file storage that is accessible over the industry-standard Service Message Block (SMB) protocol. It is built on Windows Server, delivering a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. FSx for Windows does not allow you to present S3 objects as files and does not allow you to write changed data back to S3. Therefore you cannot reference the "cold data" with quick access for reads and updates at low cost. Hence this option is not correct.

Amazon EMR - 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. EMR does not offer the same storage and processing speed as FSx for Lustre. So it is not the right fit for the given high-performance workflow scenario.

AWS Glue - AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. AWS Glue job is meant to be used for batch ETL data processing. AWS Glue does not offer the same storage and processing speed as FSx for Lustre. So it is not the right fit for the given high-performance workflow scenario.

References:

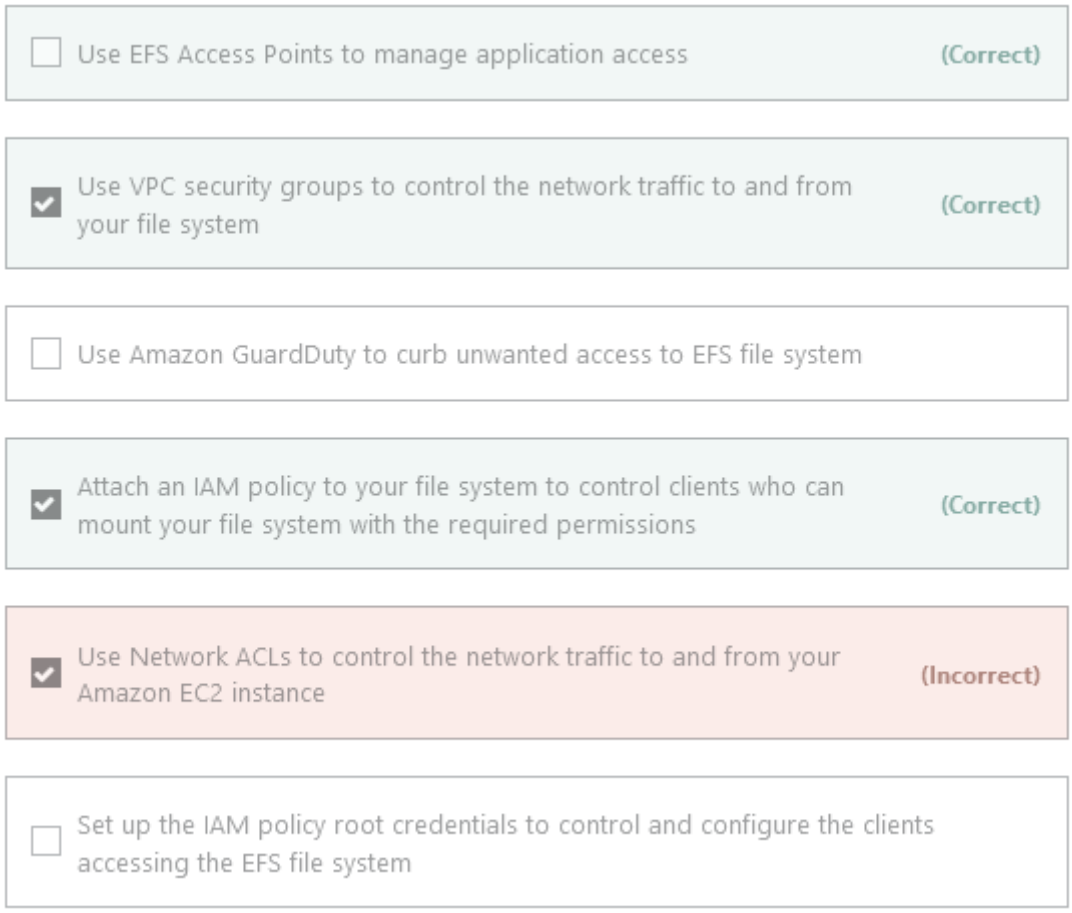
https://aws.amazon.com/fsx/lustre/

https://aws.amazon.com/fsx/windows/faqs/

1.1.17

A company has moved its business critical data to Amazon EFS file system which will be accessed by multiple EC2 instances.

As an AWS Certified Solutions Architect Associate, which of the following would you recommend to exercise access control such that only the permitted EC2 instances can read from the EFS file system? (Select three)



▶ Explanation

Correct options:

Use VPC security groups to control the network traffic to and from your file system

Attach an IAM policy to your file system to control clients who can mount your file system with the required permissions

Use EFS Access Points to manage application access

You control which EC2 instances can access your EFS file system by using VPC security group rules and AWS Identity and Access Management (IAM) policies. Use VPC security groups to control the network traffic to and from your file system. Attach an IAM policy to your file system to control which clients can mount your file system and with what permissions, and use EFS Access Points to manage application access. Control access to files and directories with POSIX-compliant user and group-level permissions.

Files and directories in an Amazon EFS file system support standard Unix-style read, write, and execute permissions based on the user ID and group IDs. When an NFS client mounts an EFS file system without using an access point, the user ID and group ID provided by the client is trusted. You can use EFS access points to override user ID and group IDs used by the NFS client. When users attempt to access files and directories, Amazon EFS checks their user IDs and group IDs to verify that each user has permission to access the objects

Incorrect options:

Use Network ACLs to control the network traffic to and from your Amazon EC2 instance - Network ACLs operate at the subnet level and not at the instance level.

Set up the IAM policy root credentials to control and configure the clients accessing the EFS file system - There is no such thing as an IAM policy root credentials and this statement has been added as a distractor.

Use Amazon GuardDuty to curb unwanted access to EFS file system - Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3. It cannot be used for access control to the EFS file system.

References:

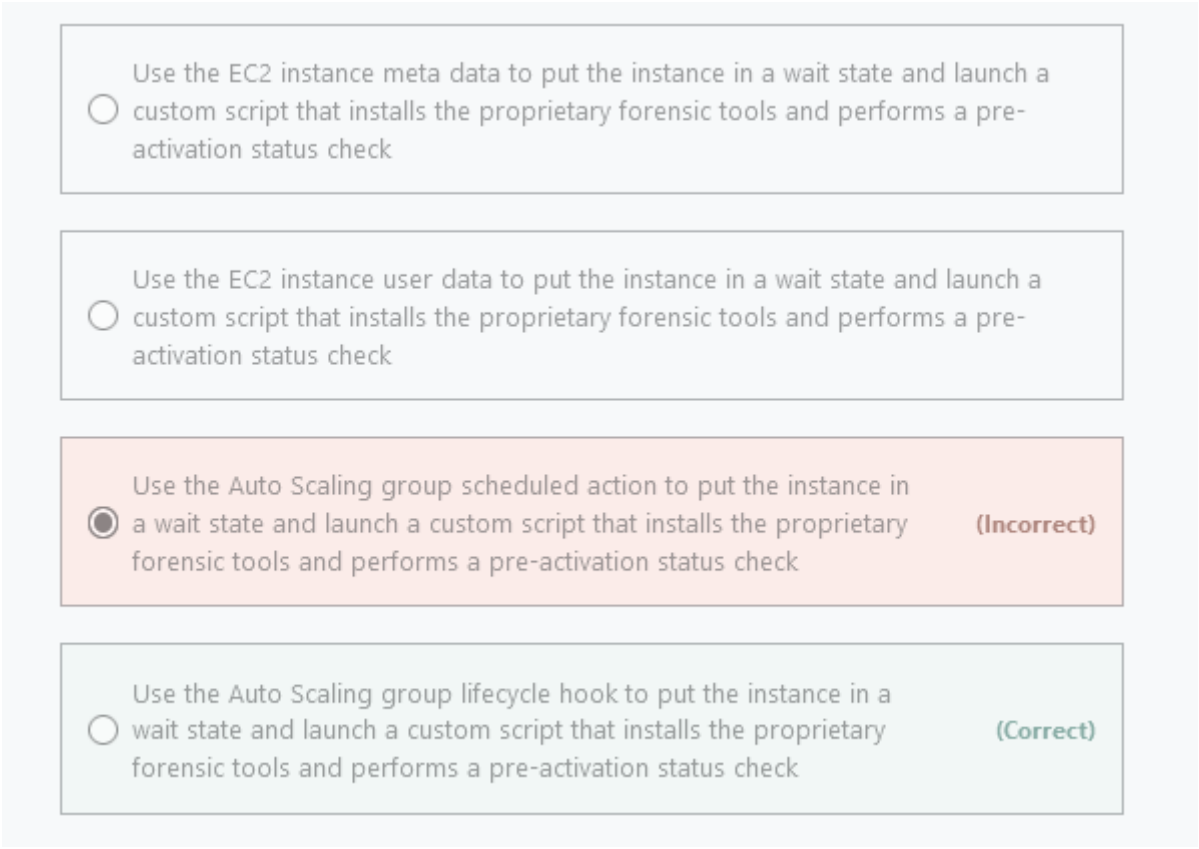
https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison

https://docs.aws.amazon.com/efs/latest/ug/accessing-fs-nfs-permissions.html

1.1.18

A company runs its EC2 servers behind an Application Load Balancer along with an Auto Scaling group. The engineers at the company want to be able to install proprietary tools on each instance and perform a pre-activation status check of these tools whenever an instance is provisioned because of a scale-out event from an auto-scaling policy.

Which of the following options can be used to enable this custom action?



▶ Explanation

Correct option:

Use the Auto Scaling group lifecycle hook to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status 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 lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances. Lifecycle hooks enable you to perform custom actions by pausing instances as an Auto Scaling group launches or terminates them. When an instance is paused, it remains in a wait state either until you complete the lifecycle action using the complete-lifecycle-action command or the CompleteLifecycleAction operation, or until the timeout period ends (one hour by default). For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates.

How lifecycle hooks work: via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html

Incorrect options:

Use the Auto Scaling group scheduled action to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. You cannot use scheduled action to carry out custom actions when the Auto Scaling group launches or terminates an instance.

Use the EC2 instance meta data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - EC2 instance metadata is data about your instance that you can use to configure or manage the running instance. You cannot use EC2 instance metadata to put the instance in wait state.

Use the EC2 instance user data to put the instance in a wait state and launch a custom script that installs the proprietary forensic tools and performs a pre-activation status check - EC2 instance user data is the data that you specified in the form of a configuration script while launching your instance. You cannot use EC2 instance user data to put the instance in wait state.

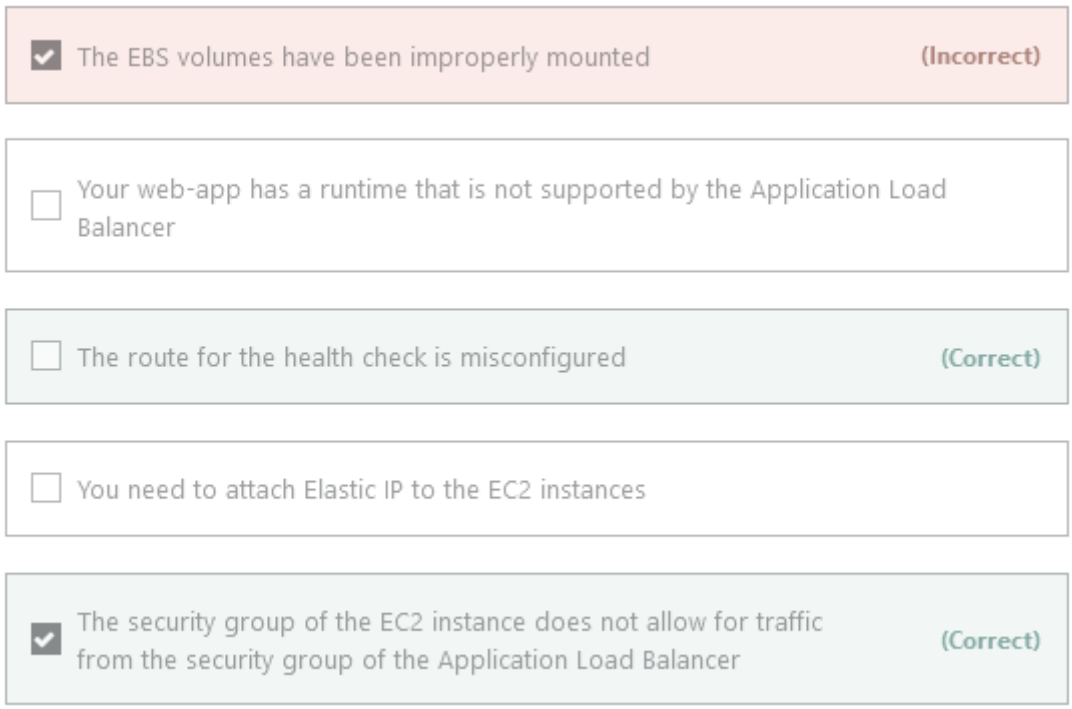
Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html

1.1.19

An Elastic Load Balancer has marked all the EC2 instances in the target group as unhealthy. Surprisingly, when a developer enters the IP address of the EC2 instances in the web browser, he can access the website.

What could be the reason the instances are being marked as unhealthy? (Select two)



▶ Explanation

Correct options

The security group of the EC2 instance does not allow for traffic from the security group of the Application Load Balancer

The route for the health check is misconfigured

An Application Load Balancer periodically sends requests to its registered targets to test their status. These tests are called health checks.

Each load balancer node routes requests only to the healthy targets in the enabled Availability Zones for the load balancer. Each load balancer node checks the health of each target, using the health check settings for the target groups with which the target is registered. If a target group contains only unhealthy registered targets, the load balancer nodes route requests across its unhealthy targets.

You must ensure that your load balancer can communicate with registered targets on both the listener port and the health check port. Whenever you add a listener to your load balancer or update the health check port for a target group used by the load balancer to route requests, you must verify that the security groups associated with the load balancer allow traffic on the new port in both directions.

Application Load Balancer Configuration for Security Groups and Health Check Routes: via - https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-update-security-

groups.html

Incorrect options:

The EBS volumes have been improperly mounted - You can access the website using the IP address which means there is no issue with the EBS volumes. So this option is not correct.

Your web-app has a runtime that is not supported by the Application Load Balancer - There is no connection between a web app runtime and the application load balancer. This option has been added as a distractor.

You need to attach Elastic IP to the EC2 instances - This option is a distractor as Elastic IPs do not need to be assigned to EC2 instances while using an Application Load Balancer.

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-update-security-groups.html

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

1.1.20

The DevOps team at an IT company is provisioning a two-tier application in a VPC with a public subnet and a private subnet. The team wants to use either a NAT instance or a NAT gateway in the public subnet to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet but needs some technical assistance in terms of the configuration options available for the NAT instance and the NAT gateway.

As a solutions architect, which of the following options would you identify as CORRECT? (Select three)

▶ Explanation

Correct options:

NAT instance can be used as a bastion server

Security Groups can be associated with a NAT instance

NAT instance supports port forwarding

A NAT instance or a NAT Gateway can be used in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the Internet.

How NAT Gateway works: via - https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html

How NAT Instance works: via - https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html

Please see this high-level summary of the differences between NAT instances and NAT gateways relevant to the options described in the question:

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

Incorrect options:

NAT gateway supports port forwarding

Security Groups can be associated with a NAT gateway

NAT gateway can be used as a bastion server

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

Reference:

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

1.1.21

The engineering team at an e-commerce company is working on cost optimizations for EC2 instances. The team wants to manage the workload using a mix of on-demand and spot instances across multiple instance types. They would like to create an Auto Scaling group with a mix of these instances.

Which of the following options would allow the engineering team to provision the instances for this use-case?

- ○ You can use a launch configuration or a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost
- ◉ You can only use a launch configuration to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost (Incorrect)
- ○ You can only use a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost (Correct)
- ○ You can neither use a launch configuration nor a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost

▶ Explanation

Correct option:

You can only use a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost

A launch template is similar to a launch configuration, in that it specifies instance configuration information such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances. Also, defining a launch template instead of a launch configuration allows you to have multiple versions of a template.

With launch templates, you can provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost. Hence this is the correct option.

Incorrect options:

You can only use a launch configuration to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost

You can use a launch configuration or a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost

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

You cannot use a launch configuration to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances. Therefore both these options are incorrect.

You can neither use a launch configuration nor a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances to achieve the desired scale, performance, and cost - You can use a launch template to provision capacity across multiple instance types using both On-Demand Instances and Spot Instances. So this option is incorrect.

References:

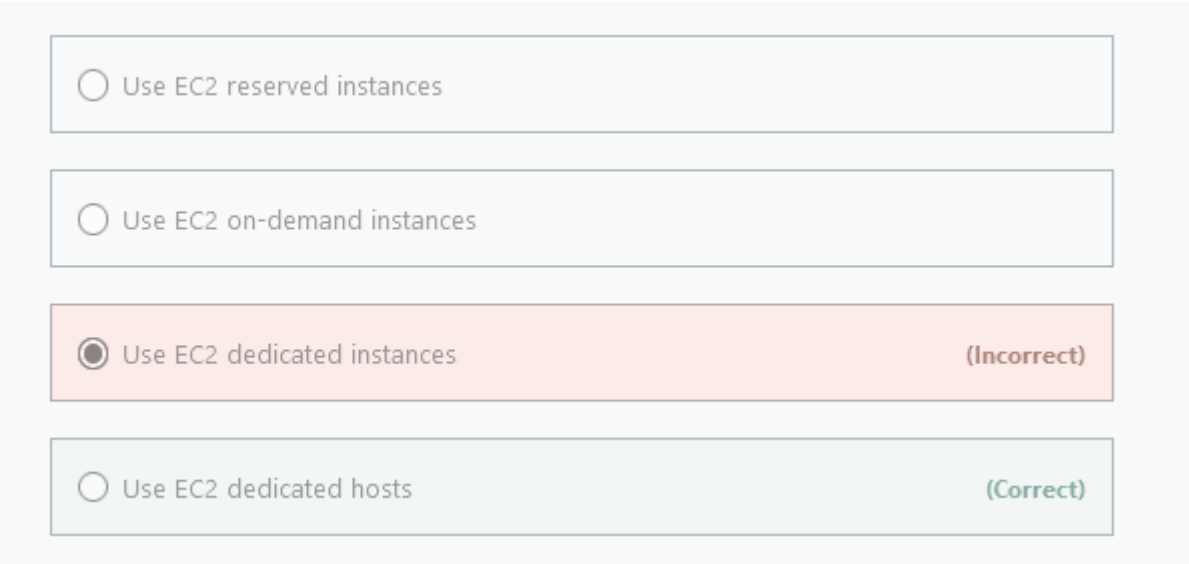
https://docs.aws.amazon.com/autoscaling/ec2/userguide/LaunchTemplates.html

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

1.1.22

A financial services company is looking to move its on-premises IT infrastructure to AWS Cloud. The company has multiple long-term server bound licenses across the application stack and the CTO wants to continue to utilize those licenses while moving to AWS.

As a solutions architect, which of the following would you recommend as the MOST cost-effective solution?



▶ Explanation

Correct option:

Use EC2 dedicated hosts

You can use Dedicated Hosts to launch Amazon EC2 instances on physical servers that are dedicated for your use. Dedicated Hosts give you additional visibility and control over how instances are placed on a physical server, and you can reliably use the same physical server over time. As a result, Dedicated Hosts enable you to use your existing server-bound software licenses like Windows Server and address corporate compliance and regulatory requirements.

Incorrect options:

Use EC2 dedicated instances - Dedicated instances are Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer. Your dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated instances may share hardware with other instances from the same AWS account that are not dedicated instances. Dedicated instances cannot be used for existing server-bound software licenses.

Use EC2 on-demand instances

Use EC2 reserved instances

Amazon EC2 presents a virtual computing environment, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network’s access permissions, and run your image using as many or few systems as you desire.

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

On-Demand Instances – Pay, by the second, for the instances that you launch.

Reserved Instances – Reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years.

Neither on-demand instances nor reserved instances can be used for existing server-bound software licenses.

References:

https://aws.amazon.com/ec2/dedicated-hosts/

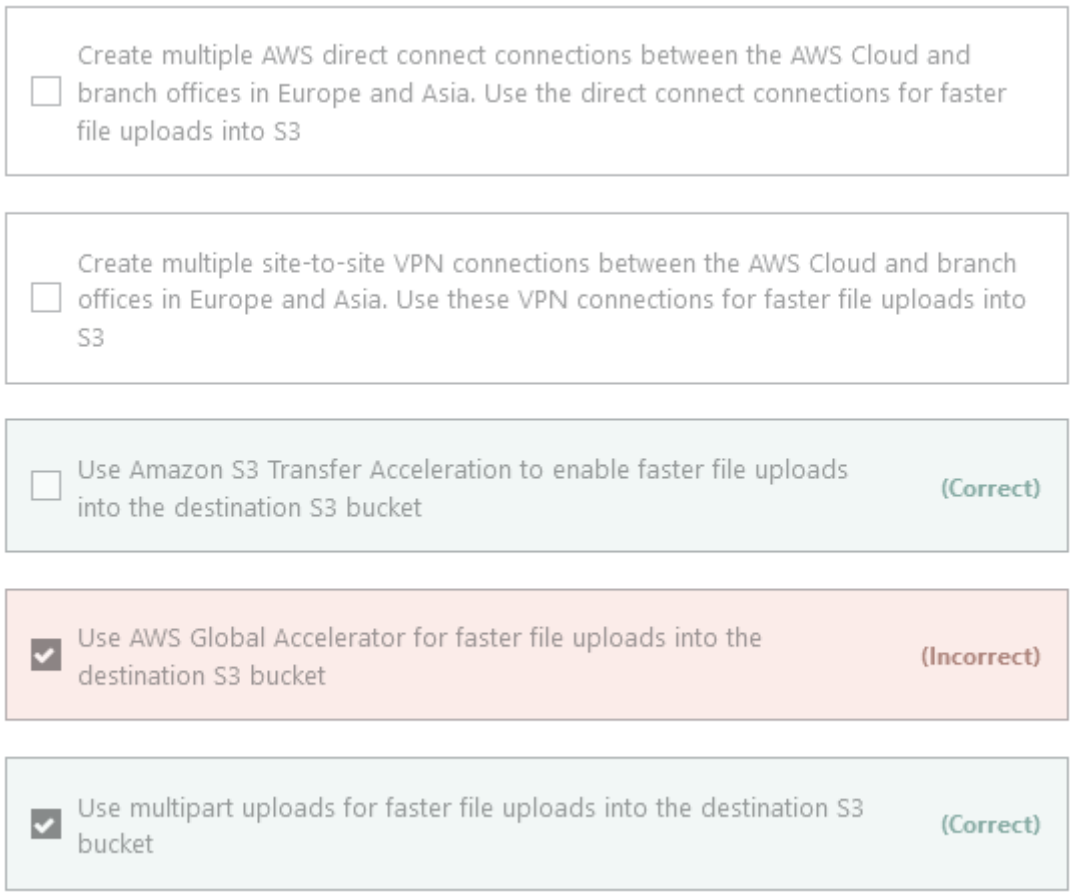
https://aws.amazon.com/ec2/dedicated-hosts/faqs/

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

1.1.23

A news network uses Amazon S3 to aggregate the raw video footage from its reporting teams across the US. The news network has recently expanded into new geographies in Europe and Asia. The technical teams at the overseas branch offices have reported huge delays in uploading large video files to the destination S3 bucket.

Which of the following are the MOST cost-effective options to improve the file upload speed into S3? (Select two)



▶ Explanation

Correct options:

Use Amazon S3 Transfer Acceleration to enable faster file uploads into the destination S3 bucket - Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. 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.

Use multipart uploads for faster file uploads into the destination S3 bucket - Multipart upload allows you to upload a single object as a set of parts. Each part is a contiguous portion of the object's data. You can upload these object parts independently and in any order. If transmission of any part fails, you can retransmit that part without affecting other parts. After all parts of your object are uploaded, Amazon S3 assembles these parts and creates the object. In general, when your object size reaches 100 MB, you should consider using

multipart uploads instead of uploading the object in a single operation. Multipart upload provides improved throughput, therefore it facilitates faster file uploads.

Incorrect options:

Create multiple AWS direct connect connections between the AWS Cloud and branch offices in Europe and Asia. Use the direct connect connections for faster file uploads into S3 - AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Direct connect takes significant time (several months) to be provisioned and is an overkill for the given use-case.

Create multiple site-to-site VPN connections between the AWS Cloud and branch offices in Europe and Asia. Use these VPN connections for faster file uploads into S3 - 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 intranet and Amazon VPC over the Internet. VPN Connections are a good solution if you have low to modest bandwidth requirements and can tolerate the inherent variability in Internet-based connectivity. Site-to-site VPN will not help in accelerating the file transfer speeds into S3 for the given use-case.

Use AWS Global Accelerator for faster file uploads into the destination S3 bucket - AWS Global Accelerator is a service that improves the availability and performance of your applications with local or global users. It provides static IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, such as your Application Load Balancers, Network Load Balancers or Amazon EC2 instances. AWS Global Accelerator will not help in accelerating the file transfer speeds into S3 for the given use-case.

References:

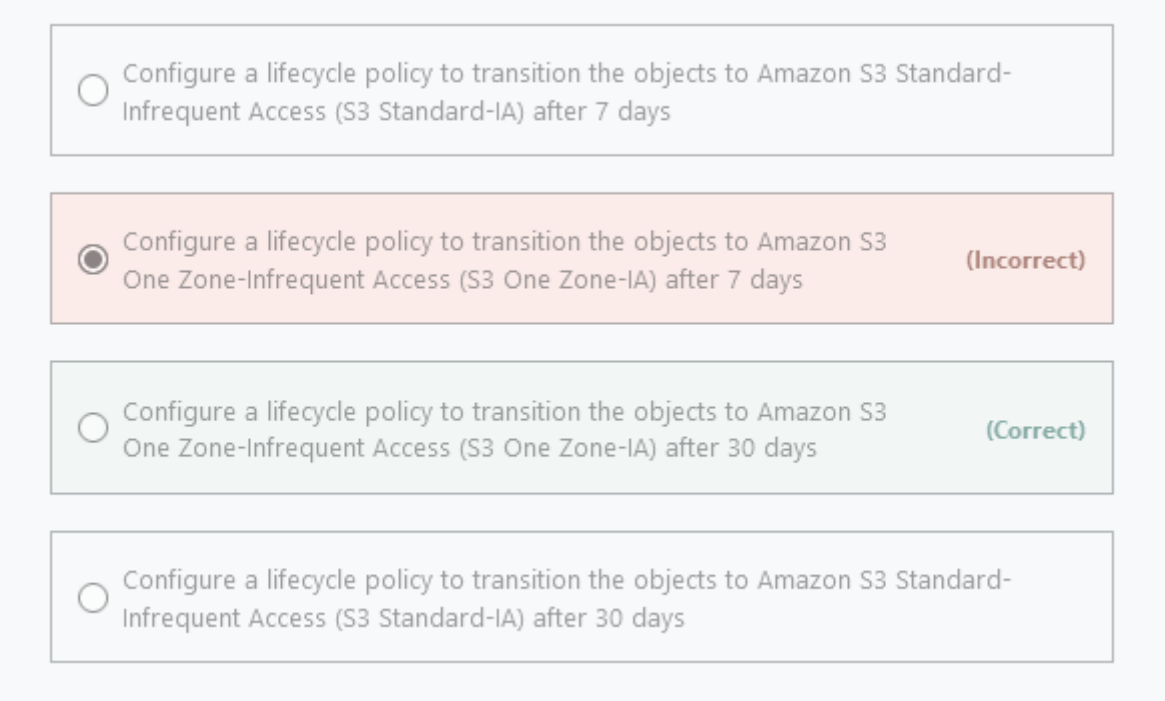
https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html

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

1.1.24

A media agency stores its re-creatable assets on Amazon S3 buckets. The assets are accessed by a large number of users for the first few days and the frequency of access falls down drastically after a week. Although the assets would be accessed occasionally after the first week, but they must continue to be immediately accessible when required. The cost of maintaining all the assets on S3 storage is turning out to be very expensive and the agency is looking at reducing costs as much as possible.

As a Solutions Architect, can you suggest a way to lower the storage costs while fulfilling the business requirements?



▶ Explanation

Correct option:

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 30 days - S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ and costs 20% less than S3 Standard-IA. S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed and re-creatable data but do not require the availability and resilience of S3 Standard or S3 Standard-IA. The minimum storage duration is 30 days before you can transition objects from S3 Standard to S3 One Zone-IA.

S3 One Zone-IA offers the same high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval fee. S3 Storage Classes can be configured at the object level, and a single bucket can contain objects stored across S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA. You can also use S3 Lifecycle policies to automatically transition objects between storage classes without any application changes.

Constraints for Lifecycle storage class transitions: via - https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html

Supported S3 lifecycle transitions: via - https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html

Incorrect options:

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 7 days

Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 7 days

As mentioned earlier, the minimum storage duration is 30 days before you can transition objects from S3 Standard to S3 One Zone-IA or S3 Standard-IA, so both these options are added as distractors.

Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 30 days - S3 Standard-IA is for data that is accessed less frequently, but requires rapid access when needed. S3 Standard-IA offers the high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval fee. This combination of low cost and high performance makes S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery files. But, it costs more than S3 One Zone-IA because of the redundant storage across availability zones. As the data is re-creatable, so you don't need to incur this additional cost.

References:

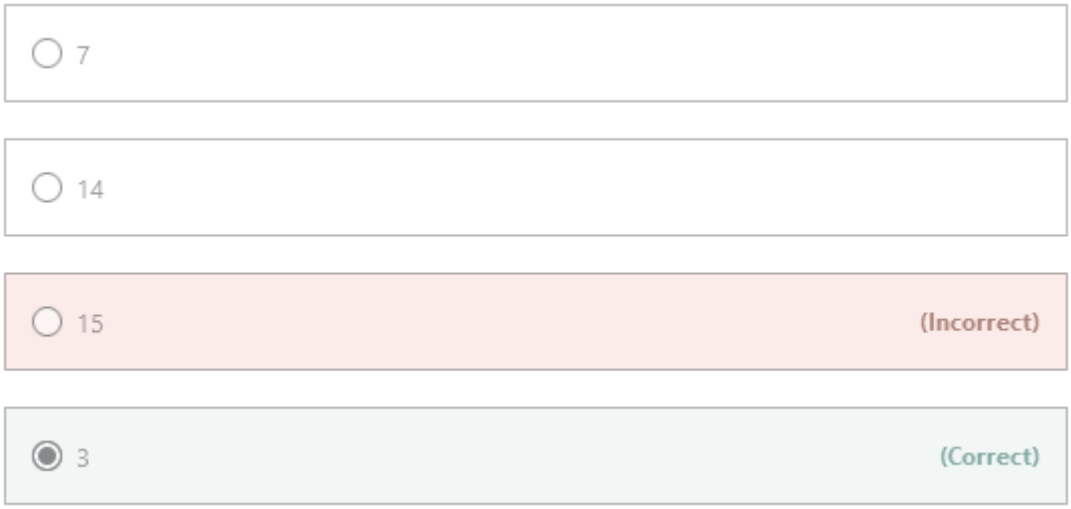
https://aws.amazon.com/s3/storage-classes/

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

1.1.25

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

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



▶ Explanation

Correct option:

3

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

Cluster 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 EC2 instances in a single Spread placement group, the company needs to use 3 AZs.

Spread placement group overview: 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