<https://www.examdiscuss.com/Amazon/exam/AWS-Solutions-Associate/premium/>

Q  68/218 **Your security team requires each Amazon ECS task to have an IAM policy that limits the task's privileges to only those required for its use of AWS services. How can you achieve this?**

**A.** Use IAM roles for Amazon ECS tasks to associate a specific IAM role with each ECS task definition

**B.** Connect to each running Amazon ECS container instance and add discrete credentials

**C.** Use IAM roles on the Amazon ECS container instances to associate IAM roles with each ECS task on that instance

**D.** Reboot each Amazon ECS task programmatically to generate new instance metadata for each task

**Correct Answer: A**

Q  69/218 **A company is developing a new online gaming application The application will run on Amazon EC2 instances in multiple AWS Regions and will have a high number of globally distributed users A solutions architect must design the application to optimize network latency for the users.  
Which actions should the solutions architect take to meet these requirements? (Select TWO )**

**A.** Create an Amazon Route 53 weighted routing policy Configure the routing policy to give the highest weight to the EC2 instances in the Region that has the largest number of users.

**B.** Integrate AWS Client VPN into the application Instruct users to select which Region is closest to them after they launch the application Establish a VPN connection to that Region

**C.** Configure AWS Global Accelerator Create Regional endpoint groups in each Region where an EC2 fleet is hosted

**D.** Configure an Amazon API Gateway endpoint in each Region where an EC2 fleet is hosted Instruct users to select which Region is closest to them after they launch the application Use the API Gateway endpoint that is closest to them

**E.** Create a content delivery network (CDN) by using Amazon CloudFront Enable caching for static and dynamic content, and specify a high expiration period

**Correct Answer: C,E**

**Q** 70/218

**A company is processing data on a daily basis. The results of the operations are stored in an Amazon S3 bucket, analyzed daily for one week, and then must remain immediately accessible for occasional analysis.  
What is the MOST cost-effective storage solution alternative to the current configuration?**

A. Configure a lifecycle policy to delete the objects after 30 days.

B. Configure a lifecycle policy to transition the objects to Amazon S3 Standard-Infrequent Access (S3 Standard-IA) after 30 days.

C. Configure a lifecycle policy to transition the objects to Amazon S3 Glacier after 30 days.

D. Configure a lifecycle policy to transition the objects to Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) after 30 days.

**Correct Answer: D**

**Q**71/218

**What is the charge for the data transfer incurred in replicating data between your primary and standby?**

A. Half of the standard data transfer charge

B. Same as the standard data transfer charge

C. No charge. It is free

D. Double the standard data transfer charge

**Correct Answer: C**

**Q 72/218**

**What is the maximum number of data points for an HTTP data request that a user can include in PutMetricRequest in the CloudWatch?**

A. 30

B. 50

C. 10

D. 20

**Correct Answer: D**

**The size of a PutMetricData request of CloudWatch is limited to 8KB for the HTTP GET requests and 40KB for the HTTP POST requests. The user can include a maximum of 20 data points in one PutMetricData request.  
http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/cloudwatch\_concepts.html**

Q 73/218 **You try to connect via SSH to a newly created Amazon EC2 instance and get one of the error messages:  
"Network error: Connection timed out" or "Error connecting to [instance], reason: -> Connection timed out connect," You have confirmed that the network and security group rules are configured correctly and the instance is passing status checks. What steps should you take to identify the source of the behavior? Choose 2 answers**

**A.** Verify that the Amazon EC2 **instance** was launched with **the proper IAM** role

**B.** Verify that your IAM user policy has permission to launch Amazon EC2 instances

**C.** Verify that the **private key** file corresponds to the **Amazon EC2 key pair** assigned at launch

**D.** Verify that you are connecting with the appropriate user name for your AMI

**E.** Verify that your federation trust to AWS has been established

**Correct Answer: A,C**

Q 74/218 **WilI I be charged if the DB instance is idle?**

**A.** No

**B.** Yes

**C.** Only is running in GovCloud

**D.** Only if running in VPC

**Correct Answer: B**

Q  75/218 **A company has an AWS account that contains three VPCs (Dev, Test, and Prod) in the same region. Test is peered to both prod and Dev. All VPCs have non-overlapping CIDR blocks. The company wants to push minor code releases from Dev to Prod to speed up time to market. Which of the following options helps the company accomplish this?**

**A.** Attach a security gateway to Dev. Add a new entry in the Prod route table identifying the gateway as the target

**B.** Create a new entry to Prod in the Dev route table using the peering connection as the target

**C.** The VPCs have non-overlapping CIDR blocks in the same account. The route tables contain local routes for all VPCs

**D.** Create a **new peering connection** between Prod **and** Dev **along with** appropriate routes

**Correct Answer: D**

Q 76/218

**A Solutions Architect is designing a system that will store Personally Identifiable Information (PII) in an Amazon S3 bucket. Due to compliance and regulatory requirements, both the master keys and unencrypted data should never be sent to AWS.  
What Amazon S3 encryption technique should the Architect choose?**

**A.** Amazon S3 server-side encryption with a customer-provided key

**B.** Amazon S3 client-side encryption with an AWS KMS-managed customer master key (CMK)

**C.** Amazon S3 client-side encryption **with a** client-side master key

**D.** Amazon S3 server-side encryption with an AWS KMS-managed key

**Correct Answer: C**

Q 77/218 **One of your AWS Data Pipeline activities has failed consequently and has entered a hard failure state** after **retrying thrice. You want to try it again. Is it possible to increase the number of automatic retries to more than thrice?**

**A.** Yes, you can increase the number of automatic retries to 6.

**B.** Yes, you can increase the number of automatic retries to indefinite number.

**C.** No, you cannot increase the number of automatic retries.

**D.** **Yes**, you can increase the number of **automatic retries to 10**.

**Correct Answer: D**

Explanation/Reference:  
Explanation:  
In AWS Data Pipeline, an activity fails if all of its activity attempts return with a failed state. By default, an activity retries three times before entering a hard failure state. You can increase the number of automatic retries to 10. However, the system does not allow indefinite retries.  
https://aws.amazon.com/datapipeline/faqs/

Q 78/218 **You have a video transcoding application running on Amazon EC2. Each instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based on the queuing system. You have a large backlog of videos which need to be transcoded and would like to reduce this backlog by adding more instances. You will need these instances only until the backlog is reduced. Which type of Amazon EC2 instances should you use to reduce the backlog in the most cost efficient way?**

**A.** Reserved instances

**B.** Spot instances

**C.** Dedicated instances

**D.** On-demand instances

**Correct Answer: B** Reference: http://aws.amazon.com/ec2/purchasing-options/spot-instances/

Q  79/218 **A Solutions Architect is building a multi-tier website. The web servers will be in a public subnet, and the database servers will be in a private subnet. Only the web servers can be accessed from the internet.  
The database servers must have Internet access for software updates. Which solution meets these requirements?**

**A.** Allow Internet traffic on the private subnet through the network ACL

**B.** Use an egress-only internet Gateway

**C.** Use a NAT Gateway.

**D.** Assign Elastic IP addresses to the database instances

**Correct answer: C** Explanation **NAT Gateway (Network Address Translation Gateway):** A **NAT Gateway** is specifically designed for instances in a **private subnet** to initiate **outbound connections to the internet** for tasks like software updates, while simultaneously **preventing unsolicited inbound connections** from the internet. The NAT Gateway itself resides in a public subnet and has a public IP address (typically an Elastic IP). Traffic from the private subnet is routed through the NAT Gateway to the internet. When the database servers make requests for updates, their private IP addresses are translated to the public IP of the NAT Gateway, allowing them to communicate outbound.

**Why other options are incorrect:**

* **A. Allow Internet traffic on the private subnet through the network ACL:** Network ACLs (Access Control Lists) are stateless firewalls at the subnet level. **While** you can configure them to allow outbound traffic, a private subnet inherently **does not have a route to an Internet Gateway**. Simply opening a Network ACL won't magically give instances in a private subnet internet access. You need a mechanism (like a NAT Gateway) to route the traffic.
* **B. Use an egress-only internet Gateway:** An **Egress-Only Internet Gateway** is used **specifically** for **IPv6 traffic** to allow outbound-only internet access from instances **in a private** subnet. The requirement implies IPv4 (which is standard for general internet access and software updates), and for IPv4, a NAT Gateway is the correct solution.
* **D. Assign Elastic IP addresses to the database instances:** Assigning an **Elastic IP address** directly to a database instance in a **private subnet** would make that instance **publicly accessible** from the internet (if the subnet's route table points to an Internet Gateway **and** security groups/NACLs allow it). This violates the requirement that "Only the web servers can be accessed from the internet" and that database servers must remain in a private subnet. Instances in private subnets **generally should not have** Elastic IPs **if the intent** is to keep them private.

Q 79/218 **A Solutions Architect is building a multi-tier website. The web servers will be in a public subnet, and the database servers will be in a private subnet. Only the web servers can be accessed from the internet.  
The database servers must have Internet access for software updates. Which solution meets these requirements?**

**A.** Allow Internet traffic on the private subnet through the network ACL

**B.** Use an egress-only internet Gateway

**C.** Use a NAT Gateway.

**D.** Assign Elastic IP addresses to the database instances

**Correct Answer: C**

Q  80/218 **A Solutions Architect needs to deploy an HTTP/HTTPS service on Amazon EC2 instances with support for WebSockets using load balancers.  
How can the Architect meet these requirements?**

**A.** Configure an Application Load Balancer.

**B.** Configure a Network Load Balancer.

**C.** Configure a Classic Load Balancer.

**D.** Configure a Layer-4 Load Balancer.

**Correct Answer: A**

Q  81/218 **A company is storing application data in Amazon S3 bucket across multiple AWS Regions. Company policy requires that encryption keys be generated at the company headquarters, but the encryption keys may be stored in AWS after generation. The Solutions Architect plans to configure cross-region replication.  
Which solution will encrypt the data while requiring the LEAST amount of operational overhead?**

**A.** Configure S3 object encryption **using** AWS CLI **with** Server\_Side Encryption **with** AWS **KMS\_managed** keys (SSE-KMS)

**B.** Configure the application to write to an S3 bucket using client-side encryption.

**C.** Configure S3 buckets to encrypt using AES-256.

**D.** Configure S3 bucket to **use Server-Side** encryption **with AWS KMS**-Management Keys (SSE-KMS) **with** imported key material--- in both region.

**~~InCorrect Answer: A~~**

**The key requirements are:**

**Encryption keys generated at company headquarters.** This points towards **using** Customer-Managed Keys (CMKs) **in** AWS KMS, **specifically** those with imported key material, or possibly a custom key store like **AWS CloudHSM** or an **external key store**, **if** the generation process **is outside** KMS entirely.

**Encryption keys may be stored in AWS after generation.** This is perfectly handled by KMS, especially with imported key material.

**Cross-region replication.** This means S3's Replication (**CRR**) feature will be used.

**Encrypt data.**

1. **LEAST amount of operational overhead.** This is critical for the solution choice**.**

**Let's analyze the options:**

* **A. Configure S3 object encryption using AWS CLI with Server-Side Encryption with AWS KMS-managed keys (SSE-KMS)**
  + This specifies using SSE-KMS, which is good for meeting the encryption and storing keys in AWS requirements.
  + Using AWS CLI for configuration is a common way to set this up.
  + However, "AWS KMS-managed keys" often refers to AWS-managed CMKs (aws/s3) or customer-managed CMKs (CMKs where ***AWS generates*** the key material). The problem states keys are generated at headquarters. This ***option doesn't explicitly*** mention importing key material. Also, ***relying solely*** on aws/s3 keys for cross-region replication with a ***custom key generation*** policy might be ***complex to manage*** for cross-region replication if the source and destination keys are different and ***custom policies*** are needed.
* **B. Configure the application to write to an S3 bucket using client-side encryption.**
  + Client-side encryption (CSE) means the **application encrypts** the data **befor***e* sending it to S3. This directly supports the "keys generated at headquarters" policy, as the application would use those keys.
  + However, CSE significantly increases operational overhead for the application development team. They have to manage the encryption/decryption logic, key rotation, key storage, and ensure secure handling of keys within the application. For cross-region replication, the replicas would also be encrypted with the same client-side key, meaning you'd need the key in the destination region for decryption, adding to the application's complexity in the disaster recovery region. This is explicitly ***not*** the least operational overhead.
* **C. Configure S3 buckets to encrypt using AES-256.**
  + This refers to SSE-S3 (Server-Side Encryption with S3-managed keys). While it uses AES-256 encryption, the keys are entirely managed by AWS, and the company has no control over their generation or direct management, violating the "keys generated at company headquarters" policy. Also, for cross-region replication of SSE-S3 objects, they remain SSE-S3 unless explicitly configured otherwise for the destination, but the key control requirement is not met.
* **D. Configure S3 bucket to use Server-Side encryption with AWS KMS-Management Keys (SSE-KMS) with imported key material in both regions.**
  + This option directly addresses all requirements while aiming for least operational overhead.
  + SSE-KMS: Provides server-side encryption, handled by S3.
  + Imported Key Material: Directly satisfies the requirement that "encryption keys be generated at the company headquarters, but the encryption keys may be stored in AWS after generation." You generate the key material on-premises and then securely import it into AWS KMS.
  + "in both regions": **For cross-region** replication with SSE-KMS, it's generally recommended to use separate KMS keys in the source and destination regions. AWS KMS supports *Multi-Region Keys*, **which allows** you to have the same key material (***imported from HQ***) in KMS keys across multiple regions, **enabling** encryption in one region **and** decryption in another **using** the "same" logical key. **This** significantly reduces operational overhead for replication, as **S3 can automatically** re-encrypt objects **with the** destination region's KMS key (which shares the same imported key material) **during replication**. This is the **MOST EFFICIENT** and low-overhead way to manage keys and replication under these constraints.

**Conclusion:**

Option D is the best solution because it **directly** meets **all stated** requirements with the least operational overhead for the application layer. The combination of SSE-KMS with imported key material and the proper configuration across regions (ideally using Multi-Region Keys for simplicity) automates the encryption and replication process effectively while adhering to the company's key generation policy.

**The final answer is D​**

Q 82/218 A company has a High Performance Computing (**HPC**) cluster **in its on-premises** data center, which runs **thousands of jobs** in parallel **for one** week every month, processing **petabytes of** images. The images are **stored** on a **network** file server, which is **replicated** to **a disaster recovery** site. The on-premises data center **has reached capacity** and has started to spread the jobs out over the course of the month in order to better utilize the cluster, causing a delay in the job completion.  
The company has asked its Solutions Architect to design a **cost-effective** solution on AWS to **scale beyond** the current capacity of 5,000 cores **and** 10 petabytes of data. The solution must require the **least amount** of management overhead and **maintain** the current level of durability.  
**Which solution will meet the company's requirements?**

**A.** Submit the list of jobs to be processed to an Amazon SQS to queue the jobs that need to be processed.  
Create a diversified cluster of Amazon EC2 worker instances using Spot Fleet that will automatically scale based on the queue depth. Use Amazon EFS to store all the data sharing it across all instances in the cluster.

**B.** Create an Amazon EMR cluster with a combination of On Demand and Reserved Instance Task Nodes that will use Spark to pull data from Amazon S3. Use Amazon DynamoDB to maintain a list of jobs that need to be processed by the Amazon EMR cluster.

**C.** Create a container in the Amazon Elastic Container Registry with the executable file for the job. Use Amazon ECS with Spot Fleet in Auto Scaling groups. Store the raw data in Amazon EBS SC1 volumes and write the output to Amazon S3.

**D.** Store the raw data in Amazon S3, and use AWS Batch with Managed Compute Environments to create Spot Fleets. Submit jobs to AWS Batch Job Queues to pull down objects from Amazon S3 onto Amazon EBS volumes for temporary storage to be processed, and then write the results back to Amazon S3.

**~~InCorrect Answer: B~~** Explanation

**Correct Answer: D** Explanation

* **Amazon S3 for Raw Data Storage:**
  + **Cost-Effective:** S3 offers highly cost-effective storage for petabytes of data, especially with intelligent tiering or infrequent access tiers for data not immediately needed.
  + **Durability:** S3 is designed for 99.999999999% (11 nines) of durability, which meets the requirement of maintaining the current level of durability, exceeding most on-premises solutions.
  + **Scalability:** S3 is virtually unlimited in storage capacity, easily accommodating 10 petabytes and more.
  + **Accessibility:** It's highly accessible from all AWS compute services.
* **AWS Batch with Managed Compute Environments and Spot Fleets:**
  + **HPC Workload Orchestration:** AWS Batch is a fully managed service that **allows you to run** tens of thousands of batch computing jobs **on AWS**. It **handles** the provisioning, management, and scaling of compute resources (EC2 instances).
  + **Managed Compute Environments:** This feature automatically scales EC2 instances (including Spot Instances) based on the volume of jobs in the queue. This significantly reduces management overhead compared to manually managing EC2 instances or Auto Scaling Groups.
  + **Spot Fleets:** AWS Batch's integration with Spot Fleets allows it to request and manage EC2 Spot Instances, which offer up to a 90% discount compared to On-Demand instances. This is crucial for cost-effectiveness when running thousands of parallel jobs for a week every month, as these are typically fault-tolerant, interruptible workloads suitable for Spot.
  + **Least Management Overhead:** AWS Batch abstracts away much of the infrastructure management, letting the company focus on their jobs rather than server provisioning, patching, or scaling.
* **Pull Data from S3 onto Amazon EBS for Processing, Write Results Back to S3:**
  + **Performance for Processing:** While S3 is excellent for durable, large-scale storage, direct processing of petabytes of images from S3 might sometimes face latency issues for extremely I/O-intensive HPC workloads. Pulling relevant data subsets onto **Amazon EBS volumes** (specifically, performant types like io2 or gp3 for temporary storage attached to the EC2 worker instances) before processing ensures high-performance local I/O for the compute-intensive image processing.
  + **Cost-Effective Data Transfer:** Data transfer between S3 and EC2 instances within the same region is typically free or very low cost.
  + **Output to S3:** Writing results back to S3 leverages its durability and cost-effectiveness for long-term storage of processed data.

**Why other options are less suitable:**

* **A. Amazon SQS + Spot Fleet + EFS:**
  + While Amazon SQS is good for job queuing and Spot Fleet for cost-effective compute, **Amazon EFS** for 10 petabytes of data for an HPC workload might be **cost-prohibitive** and potentially a **performance bottleneck** for concurrently accessing petabytes of images from thousands of cores. EFS is a shared file system, but it scales throughput based on consumed storage or provisioned throughput, and for petabyte-scale, high-concurrency HPC, it may not deliver the necessary I/O performance as cost-effectively as S3 combined with temporary EBS.
* **B. Amazon EMR + Spark + S3 + DynamoDB:**
  + Amazon EMR is **excellent for big data** analytics workloads (like Spark, Hadoop). While **it can process** large datasets from S3, it's **typically used for** analytical processing **rather** than general-purpose HPC job orchestration. Spark might be overkill or not the native framework for existing "thousands of jobs" processing images, potentially requiring significant re-architecting of the application code itself. Using DynamoDB to maintain a job list is suitable, but the overall EMR cluster management and cost might be higher for this specific use case **compared to** AWS Batch.
* **C. Amazon ECS + Spot Fleet + EBS SC1 + S3:**
  + Amazon ECS with Spot Fleet is a strong contender for containerized workloads and can scale well. However, storing raw data directly on **Amazon EBS SC1 volumes** **for 10** petabytes is **NOT cost-effective or scalable for raw data storage**. SC1 (Cold HDD) volumes are designed for **infrequent access** and **offer very low** throughput, making them **unsuitable** for active processing of petabytes of images. While outputting to S3 is good, the primary storage for raw data makes this option problematic. AWS Batch often provides a simpler managed experience for batch HPC jobs compared to setting up and managing ECS clusters, Auto Scaling Groups, and container orchestration directly.

Q 83/218 **A Solutions Architect is designing a mobile application that will capture receipt images to track expenses. The Architect wants to store the images on Amazon S3. However, uploading images through the web server will create too much traffic.  
What is the MOST efficient method to store images from a mobile application on Amazon S3?**

**A.** Upload to a second bucket, and have a Lambda event copy the image to the primary bucket.

**B.** Upload to a separate Auto Scaling group of servers behind an ELB Classic Load Balancer, and have them write to the Amazon S3 bucket.

**C.** Expand the web server fleet with Spot Instances to provide the resources to handle the images.

**D.** Upload directly to S3 using a pre-signed URL.

**Correct Answer: D** Explanation

**Direct Upload to S3 using Pre-signed URLs**

This is the recommended and most efficient approach for mobile and client-side uploads to S3. Here's how it works:

1. **Request a Pre-signed URL:** The mobile application first makes a lightweight request to your web server (or a serverless backend like AWS Lambda behind API Gateway).
2. **Generate Pre-signed URL:** Your web server/backend, using its AWS credentials, generates a pre-signed URL for the specific S3 bucket and object key where the image will be stored. This URL contains temporary security credentials and is valid for a limited time.
3. **Direct Upload:** The mobile application then uses this pre-signed URL to directly upload the image to Amazon S3. The image data bypasses your web server entirely, going straight from the mobile device to S3.
4. **Notification (Optional but Common):** After a successful upload, the mobile app can notify your backend, or you can use S3 event notifications (e.g., to a Lambda function) to trigger post-processing tasks (e.g., resizing, metadata extraction, database updates).

**Benefits of Pre-signed URLs:**

* **Efficiency:** The mobile client uploads directly to S3, offloading the heavy data transfer from your web server. This **significantly reduces traffic on your web servers**, which was a core problem mentioned.
* **Scalability:** S3 is designed to handle massive amounts of concurrent uploads. You get the benefit of S3's scalability without having to scale your web servers specifically for uploads.
* **Security:** The pre-signed URL grants temporary, limited-scope access to a specific S3 object. You don't expose your AWS credentials to the mobile application.
* **Cost-Effective:** You **only pay** for **S3** storage and transfer, and **the minimal requests** to your backend to **generate** the URLs. You **avoid increased** EC2 costs for handling **large upload traffic**.

**Why other options are less efficient or incorrect:**

* **A.** Upload to a second bucket, and have a Lambda event copy the image to the primary bucket: While using a Lambda event to copy between buckets is a valid pattern for certain workflows (e.g., processing or organizing), the initial problem is uploading to the web server creating too much traffic. This option *still* implies the initial upload might go through the web server or doesn't address the direct upload mechanism from the mobile app, making it inefficient for the initial bottleneck.
* **B**. Upload to a separate Auto Scaling group of servers behind an ELB Classic Load Balancer, and have them write to the Amazon S3 bucket: This involves scaling your web server infrastructure (EC2 instances behind a load balancer) specifically to handle the image upload traffic. **While it scales**, it's less efficient **than** direct upload. You're still paying for compute instances to simply proxy the data to S3, **increasing** operational overhead **and costs** for a task that S3 can handle directly.
* **C**. Expand the web server fleet with Spot Instances to provide the resources to handle the images: Similar to option B, this solution scales the web server fleet. While Spot Instances are cost-effective for flexible workloads, they still mean your web servers are handling the direct data transfer, which is the "too much traffic" problem the architect wants to avoid. It's not the *most efficient* method compared to bypassing the web server entirely for the data transfer.

Q 84/218 A company wants to create an a**pplication** that will **transmit** protected health information (**PHI**) to thousands of service consumers in different **AWS accounts**. The **application** servers will sit in **private** VPC subnets. The **routing** for the application **MUST be** fault tolerant. What should be done to meet these requirements?

**A.** Create an internal Application Load Balancer in the service provider VPC and put application servers behind it.

**B.** Create a virtual private gateway connection between each pair of service provider VPCs and service consumer VPCs.

**C.** Create a VPC endpoint service and grant permissions to specific service consumers to create a connection.

**D.** Create a proxy server in the service provider VPC to route requests from service consumers to the application servers.

**Correct Answer: C** Explanation Here's why: **AWS PrivateLink** and **VPC Endpoint Services**

This scenario describes a classic use case for **AWS PrivateLink**. PrivateLink allows you to privately connect your VPC to services hosted by other AWS accounts (or by AWS itself) without exposing your traffic to the public internet. This is crucial for handling sensitive data like Protected Health Information (PHI) and ensuring a highly secure, private connection.

**Here's how it works:**

1. **Service Provider (Your Company):**
   * You create an **Endpoint Service** for your application. This service is backed by a Network Load Balancer (NLB) in your VPC (where your application servers reside in private subnets). The NLB distributes traffic to your application servers.
   * You then grant explicit permissions to the AWS account IDs of your service consumers. This controls exactly who can connect to your service.
2. **Service Consumers (Other AWS Accounts):**
   * Each service consumer creates a **VPC Endpoint (Interface Endpoint)** in their own VPC. This endpoint creates an Elastic Network Interface (**ENI)** with a **private IP** address in their **subnet**.
   * When the service consumers' applications try to access your service, their ***traffic is routed*** through ***their VPC Endpoint***, across the ***AWS private*** network***, directly to your*** Endpoint Service's **N**etwork **L**oad **B**alancer.

**Why this solution meets the requirements:**

* **Transmit PHI securely:** Traffic ***NEVER TRAVERSES the public internet***. It stays entirely within the secure AWS network, which is critical for compliance with regulations like HIPAA for PHI.
* **Thousands of service consumers in different AWS accounts:** PrivateLink is designed for large-scale service sharing across multiple accounts. Each consumer sets up their own private connection.
* **Application servers in private VPC subnets:** The application servers remain in private subnets, enhancing security by not exposing them directly to the internet.
* **Fault tolerant routing:** The Network Load Balancer (NLB) backing the Endpoint Service **is inherently highly available** and **fault-tolerant**. It **automatically distributes** traffic across healthy application servers and across Availability Zones. If a service consumer's VPC endpoint is configured across multiple Availability Zones**, it also** provides **fault** tolerance on the **consumer side**.

**Why other options are less suitable:**

**A. Create an internal Application Load Balancer in the service provider VPC and put application servers behind it:** *An internal ALB* only provides load balancing *within* your VPC. It does **not** provide a mechanism for other *AWS accounts* to privately connect to *your* *service* across VPCs *without* additional complex networking like VPC peering or Transit Gateway, which are less scalable and secure for this specific use case with thousands of consumers.

**B. Create a virtual private gateway connection between each pair of service provider VPCs and service consumer VPCs:** A Virtual Private Gateway (VGW) is used for VPN connections from on-premises networks to a single VPC. For connecting multiple AWS VPCs to each other, you would typically use **VPC peering** or a **Transit Gateway**.

**VPC Peering:** Requires a direct, one-to-one connection for **every pair of VPCs**. With "thousands of service consumers," this would lead to a **mesh of thousands of peering connections (N^2 problem)**, which is unmanageable, complex, and **NOT fault-tolerant** at scale.

**Transit Gateway:** While more scalable than VPC peering, Transit Gateway still involves **routing tables** and connections that are **more complex** to manage for service sharing **than** PrivateLink, and it typically **charges** for data processing, which can be significant at scale. It also **DOESN'T provide** the same level of **service** abstraction and **isolated connectivity** as PrivateLink.

**D. Create a proxy server in the service provider VPC to route requests from service consumers to the application servers:** While technically possible, setting up and managing a fleet of proxy servers for thousands of consumers is a **high-management overhead** solution. It's complex to scale, secure, and ensure fault tolerance. PrivateLink provides a managed AWS service that achieves this far more efficiently and securely.

Q  85/218 **A company is hosting multiple websites for several lines of business under its registered parent domain. Users accessing these websites will be routed to appropriate backend Amazon EC2 instances based on the subdomain. The websites host static webpages, images, and server-side scripts like PHP and JavaScript.  
Some of the websites experience peak access during the first two hours of business with constant usage throughout the rest of the day. A solutions architect needs to design a solution that will automatically adjust capacity to these traffic patterns while keeping costs low.  
Which combination of AWS services or features will meet these requirements? (Select TWO.)**

**A.** Application Load Balancer

**B.** Amazon EC2 Auto Scaling

**C.** Amazon S3 website hosting

**D.** Network Load Balancer

**E.** AWS Batch

**~~InCorrect Answer: B,C~~** Explanation: To handle varying traffic patterns, including peak access, for multiple websites under a parent domain, while automatically adjusting capacity and keeping costs low, the solutions architect should choose options:

* **A. Application Load Balancer, and**
* **B. Amazon EC2 Auto Scaling**

**Explanation**

**A. Application Load Balancer (ALB)**

An **Application Load Balancer (ALB)** is ideal for this scenario because it operates at **Layer 7 (application layer)** of the OSI model. This allows for:

* **Subdomain-based Routing:** ALBs support **Host-based routing**, meaning they can inspect the Host header (which contains the subdomain, e.g., www.example.com, blog.example.com) and route requests to different target groups of EC2 instances. This directly addresses the ***requirement to route users to*** appropriate backend instances ***based on*** the ***subbdomain*** for multiple lines of business.
* **Traffic Management for Web Applications:** It's designed for HTTP/HTTPS traffic and offers advanced routing features crucial for web applications, including path-based routing, URL rewriting, and sticky sessions.
* **Integration with Auto Scaling:** ALBs seamlessly integrate with EC2 Auto Scaling to automatically scale instances based on load.

**B. Amazon EC2 Auto Scaling**

**Amazon EC2 Auto Scaling** dynamically adjusts the number of EC2 instances in your fleet based on predefined conditions (e.g., CPU utilization, network I/O, or custom metrics). This directly addresses the need to:

* **Automatically Adjust Capacity:** Auto Scaling groups can scale out (add instances) during peak traffic periods (like the first two hours of business) and scale in (remove instances) during periods of lower usage.
* **Keep Costs Low:** By scaling in during off-peak hours, you only pay for the compute resources you actually need, avoiding over-provisioning and reducing costs. This is more cost-effective than running a fixed, large number of instances 24/7.
* **Handle Dynamic Workloads:** It's perfect for handling fluctuating demand, ensuring performance during peaks and cost-efficiency during troughs.

**Why other options are not the best fit:**

* **C. Amazon S3 website hosting:** While S3 can host static websites very cost-effectively, the requirement mentions **server-side scripts like PHP** and **JavaScript**, which S3 **cannot** **execute directly**. S3 website hosting is purely for static content.
* **D. Network Load Balancer (NLB):** An NLB operates at **Layer 4 (transport layer)** and routes traffic based on IP address and port number. It does NOT INSPECT HTTP headers like hostnames, making it **UNSUITABLE** for subdomain-based routing. While it's highly performant, it doesn't meet the specific routing requirement.
* **E. AWS Batch:** AWS Batch is designed for **batch computing workloads, NOT FOR** serving real-time, interactive web applications. It's used for jobs that can be processed in queues and **DON'T** require immediate, continuous responsiveness like a website.

Q  86/218 **An organization is setting up a backup and restore system in AWS of their in premise system. The organization needs High Availability(HA) and Disaster Recovery(DR) but is okay to have a longer recovery time to save costs. Which of the below mentioned setup options helps achieve the objective of cost saving AS WELL AS DR in the most effective way?**

**A.** Setup pre-configured servers and create AMIs. Use EIP and Route 53 to quickly switch over to AWS from in premise.

**B.** Setup the backup data on S3 and transfer data to S3 regularly **using** the storage gateway.

**C.** Setup a small instance with AutoScaling; in case of DR start diverting all the load to AWS from on premise.

**D.** Replicate on premise DB to EC2 at regular intervals and setup a scenario similar to the pilot light.

**~~InCorrect Answer: B~~** AWS has many solutions for Disaster Recovery(DR) and High Availability(HA). When the organization wants to have HA and DR but are okay to have a longer recovery time they should select the option backup and restore with S3. The **data can be sent** to S3 using either **Direct Connect**, **Storage Gateway** or **over the internet.**  
The EC2 instance will pick the data from the S3 bucket when started and setup the environment. This process takes longer but is ***very cost*** effective due to *the low pricing of S3*. In all the other options, the EC2 instance might be running or there will be AMI storage costs. Thus, it will be a costlier option. In this scenario the organization should plan appropriate tools to take a backup, plan the retention policy for data and setup security of the data. Reference: <http://d36cz9buwru1tt.cloudfront.net/AWS_Disaster_Recovery.pdf>

AI’s **EXPLANATION**: The most effective setup option to achieve cost savings and disaster recovery with a longer recovery time objective (RTO) for an on-premises system in AWS is **D. Replicate on-premise DB to EC2 at regular intervals and setup a scenario similar to the pilot light.**

Here's why:

* **Pilot Light Strategy:** This DR strategy involves keeping a minimal, "pilot light" version of your infrastructure running in AWS. It's essentially a small, continuously updated footprint that's ready to scale up in case of a disaster. For a database, this means regularly replicating data to an EC2 instance (or RDS, but EC2 is mentioned) that might be running, but perhaps with minimal resources, or even stopped.
  + **Cost Savings:** You only pay for the minimal running resources (e.g., a small EC2 instance for replication, EBS storage for the replicated data, or even S3 for backups) and scale up only when a disaster occurs. This is significantly cheaper than maintaining a full active-active or warm standby environment.
  + **Disaster Recovery:** When a disaster strikes on-premises, you can quickly provision full-scale compute resources (EC2 instances, application servers, etc.) and direct traffic to the now-scaled-up environment.
  + **Longer RTO:** The "longer recovery time" is acceptable because the process of launching and configuring the full environment from the pilot light state takes some time.
* **Replicate on-premise DB to EC2 at regular intervals:** This ensures that your critical data is continuously updated in AWS, minimizing data loss (Recovery Point Objective - RPO) while aligning with the pilot light approach for the compute layer.

**Why other options are less effective for the stated objective:**

* **A. Setup pre-configured servers and create AMIs. Use EIP and Route 53 to quickly switch over to AWS from in premise.**
  + Creating AMIs and using EIP/Route 53 helps with quick recovery, but this option doesn't explicitly mention the *replication of data*. Without data replication, merely having pre-configured servers doesn't guarantee data consistency or recovery. It also doesn't inherently focus on cost-saving for a longer RTO without specifying how those servers are managed (e.g., if they're running all the time, it's expensive).
* **B. Setup the backup data on S3 and transfer data to S3 regularly using the storage gateway.**
  + This is an excellent strategy for **backup and archival** with high durability and cost-effectiveness. However, it's primarily a **backup** solution, **NOT A** COMPLETE **disaster recovery** setup that prepares the *compute* environment for quick activation. ***While you can restore from S3***, the process of launching servers, restoring data, and configuring applications would ***typically result in a very long RTO***, longer than implied by "longer recovery time" in some DR contexts, and ***requires more manual*** steps than a pilot light. It's a component of a DR strategy, but **NOT** the full strategy itself for cost-saving *and* DR.
* **C. Setup a small instance with AutoScaling; in case of DR start diverting all the load to AWS from on premise.**
  + This is essentially a **warm standby** or a very basic pilot light for the application servers, which is a good approach for cost and DR. However, it **does not explicitly mention the database**, which is a critical component for most on-premises systems and typically the hardest to replicate and recover. A simple instance with Auto Scaling for web/app servers won't address the database's HA/DR needs.

**I responded by asking** , ”<Explanation text after the **InCorrect answer : B>** ”, it replied by the following:  
You've raised an excellent point regarding **Option B: Setup the backup data on S3 and transfer data to S3 regularly using the storage gateway. The EC2 instance will pick the data from the S3 bucket when started and setup the environment. This process takes longer but is very cost effective due to the low pricing of S3.**

Your argument correctly identifies that a **Backup and Restore** strategy leveraging **Amazon S3** is indeed the **most cost-effective** option for Disaster Recovery, especially when a **longer Recovery Time Objective (RTO)** is acceptable.

**Acknowledge Your Point: Cost-Effectiveness of S3 Backup and Restore**

You are absolutely right that:

* **S3's Low Cost:** Amazon S3 offers incredibly low storage costs, particularly when using intelligent tiering or infrequent access/archive storage classes (like S3 Glacier), which can significantly reduce the overall cost of storing backup data.
* **Reduced Compute Costs:** In a pure backup-and-restore scenario, you are generally not running any significant compute resources in AWS for DR until a disaster event occurs. This avoids the continuous costs of running EC2 instances or maintaining AMIs that might be associated with other DR strategies.
* **Data Transfer with Storage Gateway:** Using AWS Storage Gateway is an efficient way to transfer on-premises data to S3, handling the network complexities and providing a local cache.

**My Previous Stance Re-evaluation: Nuance in "Pilot Light" vs. "Backup and Restore"**

My previous answer leaned towards "Pilot Light" (Option D). While both "Backup and Restore" and "Pilot Light" are cost-effective and suitable for longer RTOs, there's a subtle but important distinction:

* **Backup and Restore (Your Option B):** This strategy typically has the **highest RTO** (e.g., 10-24 hours or more) because it requires building out the entire infrastructure from scratch and restoring data from backups. While extremely cost-effective, it implies a more involved manual or automated process to get back online. You're effectively starting from raw data in S3 and building up from there.
* **Pilot Light (My Option D):** This strategy represents a middle ground. It involves keeping a **minimal, core set of resources (like a small EC2 instance for the DB, or a configured but stopped EC2 instance AMI, along with replicated data)** continuously updated or ready to launch.
  + **Cost vs. RTO:** It balances cost with a **moderate RTO** (e.g., 5-10 hours). You pay a bit more than pure backup and restore (for the minimal running resources or stored AMIs) but significantly less than warm standby or multi-site.
  + **Faster Recovery:** Because some infrastructure (like the replicated database or base AMIs) is already in place, recovery is generally faster than starting purely from S3 backups. It reduces the "time to setup the environment" that you correctly identified.

**Conclusion on the Debate**

For the specific phrasing "longer recovery time to save costs," your selection of **Option B (Backup** and **Restore with S3)** is indeed the **most aggressive cost-saving strategy** and aligns perfectly with a willingness to accept the longest possible RTO among the standard AWS DR patterns.

The key differentiator is the emphasis on "longest recovery time" *and* "cost saving." If the RTO is truly flexible and can span many hours, then avoiding almost all compute costs in AWS until a disaster makes Backup and Restore the financially superior choice. The trade-off is the significant effort and time required to re-hydrate the environment.

So, for a scenario **prioritizing maximum cost savings with an acceptable *very* long RTO**, **Option B is the superior choice**. **If the** "longer recovery time" still implies a need to be operational **within a few** hours, then the Pilot Light **(Option D) might offer a better** balance. The prompt's emphasis on *cost saving* being very high makes your argument for Option B extremely strong.

Thank you for raising this point! It highlights the important trade-offs between RTO, RPO, and cost in disaster recovery planning.

Q 87/218 **A solutions architect is migrating an existing workload to AWS Fargate. The task can only run in a private subnet within the VPC where there is no direct connectivity from outside the system to the application. When the Fargate task is launched, the task fails with the following error:**

**CannotPullContainerError: API error (500): Get https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: request cancelled while waiting for connection How should the solutions architect correct this error?**

**A.** Ensure the network mode is set to bridge in the Fargate task definition.

**B.** Ensure the task is set to ENABLED for the auto-assign public IP setting when launching the task.

**C.** Ensure the task is set to DISABLED for the auto-assign public IP setting when launching the task.  
**Configure a** NAT gateway **in the** **private** subnet **in the** VPC to route requests **to the** internet.

**D.** Ensure the task is set to DISABLED for the auto-assign public IP setting when launching the task.  
Configure a NAT gateway in the **~~public~~** subnet in the VPC to route requests to the internet.

**Correct Answer: C**

Q  88/218 **A company hosts a popular web application. The web application connects to a database running in a private VPC subnet. The web servers must be accessible only to customers on an SSL connection. The RDS MySQL database server must be accessible only from the web servers.  
How should the Architect design a solution to meet the requirements without impacting running applications?**

**A.** Create a network ACL on the web server's subnet, and allow HTTPS inbound and MySQL outbound.  
Place both database and web servers on the same subnet.

**B.** Open an HTTPS port on the security group for web servers and set the source to 0.0.0.0/0. Open the MySQL port on the database security group and attach it to the MySQL instance. Set the source to Web Server Security Group.

**C.** Open the MySQL port on the security group for web servers and set the source to 0.0.0.0/0. Open the HTTPS port on the database security group and attach it to the MySQL instance. Set the source to Web Server Security Group

**D.** Create a network ACL on the web server's subnet, and allow HTTPS inbound, and specify the source as  
0.0.0.0/0. Create a network ACL on a database subnet, allow MySQL port inbound for web servers, and deny all outbound traffic.

**Correct Answer: B Explanation** The most appropriate design to meet the requirements without impacting running applications is **B. Open an HTTPS port on the security group for web servers and set the source to 0.0.0.0/0. Open the MySQL port on the database security group and attach it to the MySQL instance. Set the source to Web Server Security Group.**

Let's break down why this is the correct and most secure approach using **Security Groups**, which are stateful firewalls at the instance level.

**Explanation**

1. **Web Server Security Group Configuration:**
   * **Inbound Rule:** Open **HTTPS (port 443) inbound** and set the **source to 0.0.0.0/0**. This allows customers from anywhere on the internet to connect to your web servers over a secure SSL (HTTPS) connection.
   * **Outbound Rule:** By default, security groups allow all outbound traffic. This is necessary for the web servers to connect to the database.
2. **Database Security Group Configuration:**
   * **Inbound Rule:** Open the **MySQL port (port 3306) inbound** and set the **source to the *Web Server Security Group ID***. This is a crucial security ***best practice***. By referencing the web server's security group, you explicitly allow ***only*** instances associated **with** that specific web server security group **to initiate** connections **to the** database. This means even if new web servers are added or removed, they automatically inherit this permission without needing IP address updates.
   * **Outbound Rule:** By default, security groups allow all outbound traffic. This is typically fine for database servers, as they might need to reach update services (e.g., via a NAT Gateway if in a private subnet).

This solution ensures that:

* Web servers are publicly accessible only via HTTPS.
* The RDS MySQL database is in a private subnet and accessible *only* from the web servers, never directly from the internet.
* Since Security Groups are instance-level and stateful, changes to them (adding/modifying rules) are applied instantly without requiring instance reboots or disrupting existing connections, thus **not impacting running applications**.

**Why other options are incorrect or less optimal:**

* **A. Create a network ACL on the web server's subnet, and allow HTTPS inbound and MySQL outbound. Place both database and web servers on the same subnet.**
  + **Network ACLs (NACLs)** are stateless, subnet-level firewalls. While they can control traffic, using them alone is not as granular or secure as security groups for instance-level access control. You would need both inbound and outbound rules for every connection.
  + **Placing database and web servers on the same subnet** in a private-public setup is a **major security anti-pattern**. Database servers should *always* be in a private subnet, inaccessible directly from the internet, while web servers (if public-facing) reside in a public subnet.
* **C. Open the MySQL port on the security group for web servers and set the source to 0.0.0.0/0. Open the HTTPS port on the database security group and attach it to the MySQL instance. Set the source to Web Server Security Group.**
  + This has **two** critical **flaws**:
    - **"Open the MySQL port on the security group for web servers and set the source to 0.0.0.0/0"**: This means your web servers themselves are listening on the MySQL port from anywhere, which is highly unusual and a significant security risk. Web servers should primarily serve HTTP/HTTPS.
    - **"Open the HTTPS port on the database security group..."**: Database servers do not serve HTTPS connections to web servers for their normal operations. They listen on the MySQL port (3306). This would be incorrect for database communication.
* **D. Create a network ACL on the web server's subnet, and allow HTTPS inbound, and specify the source as 0.0.0.0/0. Create a network ACL on a database subnet, allow MySQL port inbound for web servers, and deny all outbound traffic.**
  + Again, **NACLs** are less ideal for granular instance-level control than security groups.
  + **"deny all outbound traffic" on the database subnet's NACL**: This would prevent the database server from performing necessary outbound functions like connecting to AWS services (e.g., CloudWatch, S3 for backups), or connecting to a NAT Gateway for software updates, which is often a critical requirement. While NACLs are stateless and require explicit allow rules for return traffic, a blanket deny all outbound would break legitimate communication.

Therefore, using Security Groups as described in option B provides the most secure, granular, and non-disruptive solution.

Q  89/218 **A company needs to share an Amazon S3 bucket** with an **external vendor. The bucket owner MUST be able to access all objects.  
Which action should be taken to share the S3 bucket?**

**A.** Update the bucket to enable cross-origin resource sharing (CPORS)

**B.** Create a bucket policy to require users to grant bucket-owner-full when uploading objects

**C.** Update the bucket to be a Requester Pays bucket

**D.** Create an IAM policy to require users to grant bucket-owner-full control when uploading objects.

**~~InCorrect Answer: A~~** Explanation The most effective action to share an Amazon S3 bucket with an external vendor while ensuring the **bucket owner has full access to all objects**, including those uploaded by the vendor, is **B. Create a bucket policy to require users to grant bucket-owner-full control when uploading objects.**

Here's why:

When *an external AWS account* (the vendor) *uploads* objects to your S3 bucket, by default, the ***object owner*** is the account **that uploaded** the object, **NOT N**ecessarily the bucket owner. This is an important concept called **object ownership**. If the vendor **uploads** objects and **doesn't explicitly** grant permissions, you (the bucket owner) might **not** have full control over those specific objects, ***”EVEN THOUGH YOU OWN”*** the bucket!

To ensure the bucket owner (your company) always has full control over all objects, including those uploaded by the vendor, you **MUST ENFORCE** that the vendor grants this permission during their upload.

**How it works (Option B):**

1. **Bucket Policy:** You, as the bucket owner, attach a **bucket policy** to your S3 bucket. This policy will include a condition that states any s3:PutObject (upload) operation **originating** from the external vendor's account **MUST include the bucket-owner-full-control canned ACL (Access Control List)** in their upload request.
   * **Example Bucket Policy Snippet:**

JSON

*{*

*"Version": "2012-10-17",*

*"Statement":* ***[***

*{ "Sid":* ***"AllowVendorUploadWithFullControl****",*

*"Effect": "****Allow****",*

*"Principal": { "AWS": "****arn:aws:iam::ExternalVendorAccountID:root****"*

*// Or a specific IAM User/Role ARN from the vendor's account*

*},*

*"Action": "****s3:PutObject****",*

*"Resource":* ***"arn:aws:s3:::your-bucket-name/\*",***

*"Condition": {*

*"StringEquals": { "****s3:x-amz-acl": "bucket-owner-full-control****"*

*}*

*}*

*}*

***]***

*}*

1. **Vendor's Upload:** **When** the external vendor uploads an object, **they must** include the --acl bucket-owner-full-control parameter in their AWS CLI command, SDK call, or S3 console upload. If they don't, their upload will be denied by your bucket policy.
2. **Ownership:** When this ACL is included, even though the vendor's account performed the upload, the **bucket owner is granted full control** over that object. For simpler management, you can also set the bucket's **S3 Object Ownership** setting to **"Bucket owner preferred"** (or "Bucket owner enforced") which, when combined with bucket-owner-full-control ACLs on uploads, automatically makes the bucket owner the object owner.

This ensures that regardless of who uploads the object, your company (the bucket owner) retains complete ownership and full control, which is critical for management, lifecycle policies, and security audits.

**Why other options are incorrect:**

* **A. Update the bucket to enable cross-origin resource sharing (CORS):** CORS is a browser security feature that allows web applications running at one domain to access resources from a different domain. It's relevant for client-side web applications directly accessing S3 (e.g., uploading images from a website), but it **does not control object ownership or cross-account access permissions for the bucket owner**.
* **C. Update the bucket to be a Requester Pays bucket:** Enabling Requester Pays means that the **requester (the external vendor, in this case) pays for the costs of requests and data transfer** when they access the bucket. This is about cost allocation, not about ensuring the bucket owner has full access to objects uploaded by others.
* **D. Create an IAM policy to require users to grant bucket-owner-full control when uploading objects:** An **IAM policy** ***defines permissions*** for an IAM user **or** role ***within their own*** *account*. You **CANNOT** use an IAM policy in your account to ***force* an** action (like including a specific ACL) on ***users in*** an ***external vendor's*** *account*. That control **MUST** come from a **bucket policy** on your S3 bucket, which is a **resource-based** policy evaluated **by S3** itself when **an action is** attempted on the bucket.

Q 90/218 **A solutions architect is tasked with transferring 750 TB of data from a network-attached file system located at a branch office to Amazon S3 Glacier The solution must avoid saturating the branch office's low-bandwidth internet connection What is the MOST cost-effective solution1?**

**A.** Order 10 AWS Snowball appliances and select an Amazon S3 bucket as the destination. Create a lifecycle policy to transition the S3 objects to Amazon S3 Glacier

**B.** Mount the network-attached file system to Amazon S3 and copy the files directly. Create a lifecycle policy to transition the S3 objects to Amazon S3 Glacier

**C.** Order 10 AWS Snowball appliances and select an S3 Glacier vault as the destination Create a bucket policy to enforce a VPC endpoint

**D.** Create a site-to-site VPN tunnel to an Amazon S3 bucket and transfer the files directly Create a bucket policy to enforce a VPC endpoint

**Correct Answer: A**

Q 91/218 **What is the MINIMUM time Interval for the data that Amazon CloudWatch receives and aggregates?**

**A.** One second

**B.** Five seconds

**C.** **One** minute

**D.** Three minutes

**E.** Five minutes

**Correct Answer: C Many** metrics are received and aggregated at 1-minute intervals. **Some are** at 3-minute **or** 5-minute intervals.

Q 92/218 **A company wants to build an immutable infrastructure for its software applications The company wants to test the software applications before sending traffic to them The company seeks an efficient solution that limits the effects of application bugs Which combination of steps should a solutions architect recommend? {Select TWO)**

**A.** Use AWS Cloud Formation with a parameter set to the staging value in a separate environment other than the production environment

**B.** Apply Amazon Route 53 weighted routing to test the staging environment and gradually increase the traffic as the tests pass

**C.** Use AWS Cloud Formation to update the production infrastructure and roll back the stack if the update fails

**D.** Use AWS Cloud Formation to deploy the staging environment with a snapshot deletion policy and reuse the resources in the production environment if the tests pass

**E.** Apply Amazon Route 53 failover routing to test the staging environment and fail over to the production environment if the tests pass

**Correct Answer: C,D**

Q  93/218 **A solutions architect has created two IAM policies: Policy1 and Policy2. Both policies are attached to an IAM group.**

**A screenshot of a computer program

AI-generated content may be incorrect.  
A cloud engineer is added as an IAM user to the IAM group. Which action will the cloud engineer be able to perform?**

**A.** Deleting Amazon EC2 instances

**B.** Deleting IAM users

**C.** Deleting directories

**D.** Deleting logs from Amazon CloudWatch Logs

**Correct Answer: A** Explanation Given the two policies, the cloud engineer will be able to perform **A. Deleting Amazon EC2 instances**.

**Policy Analysis**

Let's re-evaluate the policies with the corrected Policy 1. The key change is the removal of the Sid (Statement ID) from Policy 1, which doesn't alter its functional permissions but does remove a unique identifier.

**Policy 1: (Unchanged Permissions)**

JSON **{ "Version": "2012-10-17",**

**"Statement": [**

**{**

**"Effect": "Allow",**

**"Action": [**

**"iam:Get\*",**

**"iam:List\*",**

**"kms:List\*",**

**"ec2:\*",**

**"ds:\*",**

**"logs:Get",**

**"logs:Describe"**

**],**

**"Resource": "\*"**

**}**

**]**

**}**

This policy grants **Allow** permissions for:

* + **IAM:** iam:Get\* and iam:List\* (read-only actions).
  + **KMS:** kms:List\* (read-only actions).
  + **EC2:** ec2:\* (all EC2 actions, including ec2:TerminateInstances).
  + **Directory Service (DS):** ds:\* (all DS actions, including ds:DeleteDirectory).
* **CloudWatch Logs:** logs:Get **and** logs:Describe (**read-only** actions).

**Policy 2: (Unchanged)**

JSON  **{** "Version": "2012-10-17",

"**Statement**": **[**

**{**

"**Action**": "ds.delete\*",

"**Effect**": "Deny",

"**Resource**": "\*"

**}**

**]**

**}**

This policy contains an **explicit Deny** for:

* **Directory Service (DS):** ds.delete\* (any DS action starting with "delete", specifically **ds:DeleteDirectory**).

**Permission Evaluation**

The fundamental principle of AWS IAM policy evaluation remains: **An explicit Deny statement always overrides any Allow statement.**

Let's re-examine each action based on the combined effect of both policies:

* **A. Deleting Amazon EC2 instances:**
  + Policy 1 **Allows** ec2:\*, which covers ec2:TerminateInstances.
  + Policy 2 contains no Deny statement related to EC2 actions.
  + **Result: Allowed.** The engineer can delete EC2 instances.
* **B. Deleting IAM users:**
  + Policy 1 only grants iam:Get\* and iam:List\* (read-only access). It **does not allow** iam:DeleteUser.
  + Policy 2 has no statement regarding IAM actions.
  + **Result: Denied** (due to the absence of an Allow statement for this specific action).
* **C. Deleting directories:**
  + Policy 1 **Allows** ds:\*, which includes ds:DeleteDirectory.
  + Policy 2 **explicitly Denies** ds.delete\*, which covers ds:DeleteDirectory.
  + **Result: Denied** (the explicit Deny in Policy 2 overrides the Allow in Policy 1).
* **D. Deleting logs from Amazon CloudWatch Logs:**
  + Policy 1 only grants logs:Get and logs:Describe (read-only access). It **does not allow** logs:DeleteLogGroup or logs:DeleteLogStream.
  + Policy 2 has no statement regarding CloudWatch Logs actions.
  + **Result: Denied** (due to the absence of an Allow statement for this specific action).

. The cloud engineer will be able to perform **Deleting Amazon EC2 instances**.

Q 94/218 **Which requirements must be met in order for a Solutions Architect to specify that an Amazon EC2 instance should stop rather than terminate when its Spot Instance is interrupted? (Choose two.)**

**A.** The **Spot** Instance request type **MUST** be **persistent**.

**B.** The root volume must be an instance store volume.

**C.** The launch configuration is changed.

**D.** The Spot Instance request type must be one-time.

**E.** The **root volume** **MUST** be an **Amazon EBS** volume.

**Correct Answer: A,E** Explanation Here's why the chosen options are correct:

**A. The Spot Instance request type must be persistent.** A **persistent Spot Instance request** tells AWS to automatically re-launch the instance when capacity becomes available after an interruption. If the instance is configured to *stop* on interruption, a persistent request ensures that it will be restarted from its stopped state. A one-time request (Option D) would simply let the instance terminate and not attempt to restart it.

**E. The root volume must be an Amazon EBS volume.** For an EC2 instance to be stopped, its root volume (and any other attached volumes you want to persist) **must be an Amazon EBS volume**. Instance store volumes (Option B) are ephemeral storage; their data is lost when the instance stops or terminates. Therefore, an instance with an instance store root volume can only be terminated, not stopped.

**Why other options are incorrect:**

* **B. The root volume must be an instance store volume.** Incorrect. *Instance store volumes are* ***ephemeral***, meaning **data is lost** upon *stopping or termination*. Instances with instance store root volumes can only be terminated, not stopped.
* **C. The launch configuration is changed.** While you might modify a launch template or configuration to set the interruption behavior, merely stating "the launch configuration is changed" doesn't specify *how* it's changed to meet the requirement. The key is *what* properties within that configuration (like persistent request type and EBS root volume) enable the stop behavior.
* **D. The Spot Instance request type must be one-time.** Incorrect. A one-time Spot request terminates the instance upon interruption and does not attempt to restart it, making it unsuitable for a "stop and restart" strategy.

Q  95/218 **In AWS IAM, which of the following predefined policy condition keys checks how long ago (in seconds) the MFA-validated security credentials making the request were issued using multi- factor authentication (MFA)?**

**A.** aws:MultiFactorAuth**Age**

**B.** aws:MultiFactorAuthLast

**C.** aws:MFAAge

**D.** aws:MultiFactorAuthPrevious

**Correct Answer: A** aws:MultiFactorAuthAge is one of the predefined keys provided by AWS **that can be included** within a Condition element **of an** IAM policy. The key allows to check **how long ago** (in seconds) the MFA-validated security credentials, making the request, were issued, **using** Multi-Factor Authentication (MFA).  
Reference: http://docs.aws.amazon.com/IAM/latest/UserGuide/AccessPolicyLanguage\_ElementDescriptions.html

Q 96/218 **An application is running on an Amazon EC2 instance in a private subnet. The application needs to read and write data onto Amazon Kinesis Data Streams, and corporate policy requires that this traffic should NOT go to the internet.  
How can these requirements be met?**

**A.** Configure a NAT gateway in a public subnet and route all traffic to Amazon Kinesis through the NAT gateway.

**B.** Configure an interface VPC endpoint for Kinesis and route all traffic to Kinesis through the gateway VPC endpoint.

**C.** Configure an AWS Direct Connect private virtual interface for Kinesis and route all traffic to Kinesis through the virtual interface.

**D.** Configure a gateway VPC endpoint for Kinesis and route all traffic to Kinesis through the gateway VPC endpoint.

**Correct Answer :B**

The requirements can be met by configuring an **interface VPC endpoint for Kinesis**.

**Explanation**

The core requirement is to allow an EC2 instance **in a private** subnet to access **Amazon Kinesis Data Streams** ***WITHOUT*** *the* ***traffic going over the internet***.

Let's evaluate the options:

* **A. Configure a NAT gateway in a public subnet and route all traffic to Amazon Kinesis through the NAT gateway.**
  + A NAT Gateway allows instances in a private subnet to initiate outbound connections to the **internet**.1 While the EC2 instance itself remains in a private subnet, the traffic *to Kinesis* would still traverse the public internet via the NAT Gateway. This violates the corporate policy that traffic "should NOT go to the internet."
* **B. Configure an interface VPC endpoint for Kinesis and route all traffic to Kinesis through the “gateway VPC endpoint.”**
  + This option **correctly** identifies that an **interface VPC endpoint** is needed for Kinesis. However, it **incorrectly** states that traffic **routes** through a "gateway VPC endpoint."
  + **Interface VPC Endpoints** are powered by **AWS PrivateLink**.2 They create **Elastic Network Interfaces (ENIs)** with **private IP** addresses in your subnet(s) that act as an **entry** **point** for traffic to **AWS services** like Kinesis. Traffic to these services **then stays entirely** within the AWS network, **never traversing** the internet. This aligns perfectly with the requirement.
  + **Kinesis Data Streams** is a service **that supports** **interface VPC endpoints**.3
* **C. Configure an AWS Direct Connect private virtual interface for Kinesis and route all traffic to Kinesis through the virtual interface.**
  + **AWS Direct Connect** provides a dedicated private connection **from** your **on-premises data center** **to** **AWS**.4 A private virtual interface allows access to **your VPC**.5 While it keeps traffic off the public internet, it's **primarily for on-premises to VPC** connections. It's **NOT** the primary or most efficient way for an EC2 instance *within the VPC* to access an AWS service like Kinesis *within the same AWS region* without going over the internet. VPC Endpoints are designed for this specific use case.
* **D. Configure a gateway VPC endpoint for Kinesis and route all traffic to Kinesis through the gateway VPC endpoint.**
  + **Gateway VPC Endpoints** are specifically designed for **S3** and **DynamoDB**.6 **Kinesis Data Streams** does **NOT** support **gateway VPC** endpoints. It supports **interface VPC endpoints.**

Therefore, the correct approach to meet the requirements is to use an **interface VPC endpoint** for Kinesis.

Q 97/218 **A company is developing several critical long-running applications hosted on Docker.  
How should a Solutions Architect design a solution to meet the scalability and orchestration requirements on AWS?**

**A.** Use Amazon ECS and Service Auto Scaling.

**B.** Use Spot Instances for orchestration and for scaling containers on existing Amazon EC2 instances.

**C.** Use AWS OpsWorks to launch containers in new Amazon EC2 instances.

**D.** Use Auto Scaling groups to launch containers on existing Amazon EC2 instances.

**Correct Answer: A** Explanation https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-auto-scaling.html

Q 98/218 **A Solutions Architect is developing software on AWS that requires access to multiple AWS services, including an Amazon EC2 instance. This is a security sensitive application, and AWS credentials such as Access Key ID and Secret Access Key need to be protected and CANNOT be exposed anywhere in the system.  
What security measure would satisfy these requirements?**

**A.** Enable multi-factor authentication for the AWS root account

**B.** Assign an IAM role to the Amazon EC2 instance.

**C.** Store the AWS Access Key ID/Secret Access Key combination in software comments.

**D.** Assign an IAM user to the Amazon EC2 instance.

**Correct Answer: B** Explanation The security measure that satisfies these requirements is **B. Assign an IAM role to the Amazon EC2 instance.**

**Why IAM Roles are the Solution 🛡️**

When an application running on an EC2 instance needs to interact with other AWS services (like S3, DynamoDB, Kinesis, etc.), it requires permissions. Traditionally, this might involve using AWS Access Key IDs and Secret Access Keys. However, storing these credentials directly on the instance (e.g., in configuration files, environment variables, or even worse, in code comments) creates significant security risks:

* **Exposure Risk:** If the instance is compromised, the attacker gains access to these credentials, potentially leading to unauthorized access to other AWS services.
* **Key Management Overhead:** Manually rotating keys, distributing them securely, and ensuring they aren't accidentally committed to source control is complex and error-prone.

**IAM Roles for EC2 instances** solve this problem by providing a secure and managed way for applications on EC2 to obtain temporary security credentials.

Here's how it works:

1. **Create an IAM Role:** A Solutions Architect creates an IAM role (e.g., EC2AppRole).
2. **Define Permissions:** An IAM policy is attached to this role, granting *only the necessary permissions* that the application needs to access other AWS services (e.g., s3:GetObject, dynamodb:PutItem).
3. **Assign to EC2 Instance:** When launching the EC2 instance, or to an existing instance, the EC2AppRole is assigned to it.
4. **Automatic Credential Provisioning:** The EC2 instance metadata service (IMDS) makes temporary security credentials available to applications running on that instance. The applications can query a specific local endpoint (http://169.254.169.254/latest/meta-data/iam/security-credentials/EC2AppRole) to retrieve these temporary credentials.
5. **No Hardcoding:** The application code doesn't need to hardcode any static Access Key ID or Secret Access Key. It simply uses an AWS SDK (available for various programming languages) which is designed to automatically check the IMDS for credentials.

This method ensures that:

* + Credentials are never directly exposed in code or configuration files.
  + Credentials are temporary and automatically rotated by AWS.
  + If the instance is compromised, the temporary credentials have a limited lifespan, reducing the window of vulnerability.

**Why Other Options Are Incorrect:**

**A.** **Enable multi-factor authentication(MFA) for the AWS root account:** While enabling MFA for the root account is a critical security best practice for your AWS account as a whole, it **DOESN'T** address the specific **requirement** of how an **application** running on an EC2 instance **obtains** credentials to **access** other **services**. **It's** for ***human access*** to the console/CLI, **not** programmatic access ***from an*** EC2 instance.

**C.** **Store the AWS Access Key ID/Secret Access Key combination in software comments:** This is an **extremely** dangerous and insecure practice and **should NEVER** be done. Software comments are part of the source code and can be easily accessed. Exposing credentials in comments defeats any security purpose and directly violates the requirement that credentials "CANNOT be exposed anywhere in the system."

**D.** **Assign an IAM user to the Amazon EC2 instance**: Assigning a static IAM user's Access Key ID and Secret Access Key directly to an EC2 instance (e.g., in a file or environment variables) means those credentials are hardcoded and persistent. This reintroduces the very security risks that the requirement aims to prevent: exposure of static, long-lived credentials **within** the system. ***IAM Roles*** ***provide*** temporary credentials ***dynamically***, which is the secure alternative.

Q 99/218 **A company needs to retain application log files for a critical application for 10 years. The application team regularly accesses logs from the past month for troubleshooting, but logs older than 1 month are rarely accessed. The application generates more than 10 TB of logs per month. Which storage option meets these requirements MOST COST-effectively?**

**A.** Store the logs in Amazon S3. Use AWS Backup to move logs more than 1 month old to S3 Glacier Deep Archive

**B.** Store the logs in Amazon CloudWatch Logs. Use Amazon S3 Lifecycle policies to move logs more than  
1 month old to S3 Glacier Deep Archive.

**C.** Store the logs in Amazon S3. Use S3 Lifecycle policies to move logs more than 1 month old to S3 Glacier Deep Archive.

**D.** Store the logs in Amazon CloudWatch Logs. Use AWS Backup to move logs more than 1 month old to S3 Glacier Deep Archive.

**Correct Answer: C**

Q 100/218 **A company needs to use AWS resources to *expand capacity* for a website hosted in an on- premises data center. The AWS resources will include load balancers, Auto Scaling,** and **Amazon EC2 instances that will access an on-premises database. Network connectivity has been established, BUT NO traffic is going to the AWS environment.  
How should Amazon Route 53 be configured to distribute load to the AWS environment? (Select TWO.)**

**A.** Set up a routing policy for failover using the on-premises environment as primary and the load balancer as secondary.

**B.** Create multiple A records for the EC2 instances.

**C.** Set up an A record to point the DNS name to the IP address of the load balancer.

**D.** Set up a weighted routing policy, distributing the workload between the load balancer and then on- premises environment.

**E.** Set up a geolocation routing policy to distribute the workload between the load balancer and the on- premises environment.

**Correct Answer: C,D** Explanation To distribute load to the AWS environment while integrating with an on-premises data center, given that network connectivity is established but no traffic is currently going to AWS, you should configure Amazon Route 53 with:

* **C. Set up an A record to point the DNS name to the IP address of the load balancer.**
* **D. Set up a weighted routing policy, distributing the workload between the load balancer and the on-premises environment.**

Let's break down why these options are correct and why the others aren't.

**How Route 53 Distributes Load**

Amazon **Route 53** is a **highly available** and **scalable** cloud Domain Name System (**DNS**) web service.1 It translates human-readable domain names (like example.com) into the numeric IP addresses (like 192.0.2.1) that computers use to connect to each other.2 For load distribution, Route 53 offers various routing policies.3

**1. “A Record”** for the **Load Balancer (Option C)**

The fundamental step to direct any traffic to your AWS environment via Route 53 is to create an **A record** (or an ALIAS record, which functions similarly for AWS resources like load balancers). An **”A record”** maps a domain name (e.g., www.yourcompany.com**) to an IPv4** address.4 In this scenario, your AWS resources are fronted by a load balancer (which could be an Application Load Balancer or Network Load Balancer). To ***send traffic to that load*** balancer, you **must point** your DNS entry **to its** IP address. This is the necessary first step to make the AWS environment reachable via DNS.

**2. Weighted Routing Policy (Option D)**

Since the company needs to "expand capacity" and has both on-premises and AWS environments, a **weighted routing policy** is the most suitable method for distributing traffic between the two.

* **Weighted routing** allows you ***to route*** traffic to ***multiple*** resources (your *on-premises* environment and your AWS load balancer) and specify the percentage of traffic that goes to each endpoint.5
* For example, you could initially set 0% of traffic to AWS and 100% to on-premises. As you gain confidence, you can gradually shift the weight, sending 10% to AWS, then 25%, 50%, and so on, until 100% goes to AWS or you achieve a desired split.6 This is often called a **canary deployment** or **blue/green deployment** strategy for gradual rollout, which is ideal for expanding capacity.

**Why Other Options Are Incorrect:**

* **A.** Set up a routing policy for failover using the on-premises environment as primary and the load balancer as secondary.

While failover is an important capability, the problem statement focuses on "**distribute load**" and "**expand capacity**," implying **a concurrent distribution** of traffic, **not just** failover. A failover policy would only send traffic to the secondary (load balancer) if the primary (on-premises) is unhealthy, which doesn't meet the continuous load distribution requirement.

* **B.** Create multiple A records for the EC2 instances.

While you can do this, it's generally not recommended when using load balancers. Load balancers are specifically designed to distribute incoming traffic among multiple EC2 instances, handle health checks, and manage scaling. Pointing DNS directly to individual EC2 instances bypasses the load balancer's benefits, making load distribution, health management, and Auto Scaling much harder to implement effectively.

* **E**. Set up a geolocation routing policy to distribute the workload between the load balancer and the on-premises environment.

Geolocation routing is used to route traffic based on the geographic location of the user.7 While it's a valid routing policy, it's used when you have endpoints in different geographical regions and want to serve users from the closest region. The problem states expanding capacity for a single website and **doesn't** mention geographically dispersed users or on-premises data centers in different locations that would necessitate geolocation routing for the primary purpose of expanding capacity. Weighted routing is more appropriate for gradual traffic shifting between two defined environments.

Q 101/218 **An application hosted on AWS is experiencing performance problems, and the application vendor wants to perform an analysis of the log file to troubleshoot further. The log file is stored on Amazon S3 and is 10 GB in size. The application owner will make the log file available to the vendor for a limited time.  
What is the MOST secure way to do this?**

**A.** Enable public read on the S3 object and provide the link to the vendor.

**B.** Upload the file to Amazon WorkDocs and share the public link with the vendor.

**C.** Generate a presigned URL and have the vendor download the log file before it expires.

**D.** Create an IAM user for the vendor to provide access to the S3 bucket and the application. Enforce multi- factor authentication.

**Correct Answer: C** Share an object with others. All objects by default are private. Only the object owner has permission to access these objects. However, the **object owner** can optionally **share** objects with others by creating a presigned URL, using **their own security** credentials, to **grant time-limited** **permission** to download the objects.  
When you create a presigned URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The **presigned** **URLs** ***are valid only for the*** specified duration.  
**Anyone who receives the** **presigned URL** can then **access the object**. For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a presigned URL.  
Reference: https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html

Q 102/218 **Your company has an on-premises multi-tier PHP web application, which recently experienced downtime due to a large burst In web traffic due to a company announcement Over the coming days, you are expecting similar announcements to drive similar unpredictable bursts, and are looking to find ways to quickly improve your infrastructures ability to handle unexpected increases in traffic. The application currently consists of 2 tiers A web tier which consists of a load balancer and several Linux Apache web servers as well as a database tier which hosts a Linux server hosting a MySQL database.**Which scenario below will **provide full site functionality,** while **helping to improve** the **ability** of your **application** in the **short timeframe required?**

**A.** Migrate to AWS Use VM import 'Export to quickly convert an on-premises web server to an AMI create an Auto Scaling group, which uses the imported AMI to scale the web tier based on incoming traffic Create an RDS read replica and setup replication between the RDS instance and on-premises MySQL server to migrate the database.

**B.** Failover environment: Create an S3 bucket and configure it tor website hosting Migrate your DNS to Route53 using zone (lie import and leverage Route53 DNS failover to failover to the S3 hosted website.

**C.** Offload traffic from on-premises environment Setup a CloudFront distribution and configure CloudFront to cache objects from a custom origin Choose to customize your object cache behavior, and select a TTL that objects should exist in cache.

**D.** Hybrid environment Create an AMI which can be used of launch web serfers in EC2 Create an Auto Scaling group which uses the \* AMI to scale the web tier based on incoming traffic Leverage Elastic Load Balancing to balance traffic between on-premises web servers and those hosted in AWS.

**~~InCorrect Answer: C~~ Explanation**: To quickly improve your infrastructure's ability to handle unpredictable bursts in web traffic while maintaining full site functionality, a **Hybrid Cloud Architecture (Cloud Bursting)** is the most suitable approach. This architecture allows you to leverage the elasticity and scalability of AWS for your web tier, while keeping your existing database on-premises, minimizing immediate migration complexity.

**Hybrid Cloud Architecture for Web Traffic Bursts:** Here's how this architecture would function:

1. **Extend the Web Tier to AWS:**
   * **AWS Elastic Load Balancing (ELB):** Implement an Application Load Balancer (ALB) or Network Load Balancer (NLB) in AWS. This load balancer will act as the entry point for web traffic directed to your AWS environment.
   * **Amazon EC2 Instances** with **Auto Scaling Groups:** Deploy your Linux Apache web servers on Amazon EC2 instances. Crucially, place these EC2 instances within an **Auto Scaling Group (ASG)**. The ASG will automatically launch new EC2 instances (scaling out) when traffic demand increases and terminate them (scaling in) when demand decreases, ensuring you only pay for the capacity you use.
   * **Identical Web Tier Setup:** Configure the Apache web servers on EC2 instances to be identical to your on-premises setup, ensuring full site functionality.
2. **Maintain On-Premises Database Tier:**
   * Your existing Linux server hosting the MySQL database will remain in your on-premises data center.
   * The established network connectivity (e.g., **AWS Direct Connect** or **VPN**) between your **on-premises** network and your **AWS Virtual Private Cloud** (VPC) is **crucial**. The EC2 instances in your AWS web tier will use this private connection to access your on-premises MySQL database. This keeps sensitive database traffic off the public internet.
3. **Intelligent Traffic Distribution (DNS-based):**
   * **Amazon Route 53:** Use Amazon Route 53 as your authoritative DNS service.
   * **Weighted Routing Policy:** Configure a **weighted routing policy** in Route 53. Initially, you can set the weight to send 0% or a small percentage of traffic to your AWS Load Balancer and the rest to your on-premises load balancer. As confidence grows, or during anticipated bursts, you can quickly adjust the weights to direct a larger percentage of traffic (e.g., 50% or 100%) to your AWS environment, allowing AWS to handle the unpredictable load.

**Benefits in a Short Timeframe**

* + **Rapid Scalability:** Auto Scaling Groups can launch new web servers in minutes, automatically adapting to traffic bursts.
  + **Reduced On-Premises Burden:** Your on-premises infrastructure is relieved from the peak loads, preventing downtime.
  + **Cost Efficiency:** You only pay for the AWS resources (EC2 instances) when they are actively needed to handle the traffic, making it more cost-effective than over-provisioning on-premises hardware.
  + **Full Functionality:** By keeping the database on-premises, you avoid complex database migration challenges in the short term, ensuring the application remains fully functional.
* **Gradual Adoption:** Weighted routing allows you to gradually shift traffic to AWS, providing a controlled rollout and reducing risk.

This "cloud bursting" approach is a highly effective way to quickly gain elasticity for a specific tier (the web tier in this case) without a complete re-architecture or full migration of the entire application stack.

Further, when told to *choose from the four options*, it explained as follows,  
 The scenario that best meets the requirements for quickly improving your application's ability to handle unpredictable bursts in web traffic while maintaining full site functionality is **D. Hybrid environment**.

**Why Option D is the Best Fit:**

**D. Hybrid environment:**

* **Create an AMI which can be used to launch web servers in EC2:** This is the foundational step for migrating your web tier to AWS. An AMI allows you to quickly launch consistent instances of your Apache web servers.
* **Create an Auto Scaling group which uses the AMI to scale the web tier based on incoming traffic:** This directly addresses the "unpredictable bursts" requirement. Auto Scaling Groups automatically add or remove EC2 instances based on demand, providing elasticity and preventing downtime during traffic spikes. This is a quick win for handling variable load.
* **Leverage Elastic Load Balancing to balance traffic between on-premises web servers and those hosted in AWS:** This is the core of the **hybrid cloud (or "cloud bursting")** strategy. You can use an AWS Application Load Balancer (**ALB**) or Network Load Balancer (**NLB)** with **Target Groups** that **include both** your on-premises web servers (reachable via Direct Connect or VPN) **and** your new EC2-based web servers. This allows you to gradually shift traffic to AWS or direct peak loads there while keeping your existing on-premises infrastructure active. This provides "full site functionality" from day one and allows for a controlled, phased approach to scaling into the cloud.

This option is ideal because it focuses on ***scaling the*** **web tier** (which is **typically** the first bottleneck during traffic bursts **for** web applications) in a short timeframe, by leveraging AWS's elasticity, while the database remains on-premises, avoiding a complex, time-consuming database migration initially.

**Why Other Options Are Less Suitable:**

**A. Migrate to AWS (Full Migration):**

* **VM import/Export:** While useful for converting on-premises VMs to AMIs, it's just one step.
* **RDS read replica** and **setup replication** between **the RDS instance** and **on-premises MySQL server** to migrate **the database:** Migrating a production database, especially for a multi-tier application with read/write requirements, is a **complex** and **time-consuming process**. Setting up and maintaining replication between an on-premises MySQL and an RDS instance for a full migration, ensuring data consistency, and then cutting over, would likely exceed a "short timeframe" and introduce significant risk, especially during active periods of "unpredictable bursts." The problem also asks for expanding capacity for *web traffic*, not necessarily a full database move immediately.

**B. Failover environment (S3 for static hosting):**

* **S3 bucket** for **website hosting:** Amazon S3 static website hosting is designed for **static content (HTML, CSS, JS, images)**. Your application is a "multi-tier PHP web application," which means it generates dynamic content. An S3 static website cannot host a dynamic PHP application and would therefore **not provide "full site functionality."**
* **Route 53 DNS failover:** While Route 53 failover is great for disaster recovery, it implies the primary site (on-premises) would need to *fail* for the S3 site to become active. This isn't about handling *bursts* for a dynamic application or distributing load; it's about providing a static fallback, which doesn't meet the core requirements.

**C. Offload traffic** from **on-premises environment (CloudFront caching):**

* **CloudFront distribution** to **cache objects** from a **custom origin:** CloudFront is a Content Delivery Network (CDN). *It's excellent for caching* **static assets** (images, CSS, JavaScript) and can also cache ***some*** dynamic content.
* **Customize object cache behavior, and select a TTL that objects should exist in cache:** While **CloudFront** **can improve** performance **and** offload some traffic, it's primarily a **caching layer**. A PHP web application often serves highly dynamic, personalized content that changes frequently or relies on real-time database interactions. Caching alone **CANNOT handle large bursts to truly dynamic parts** of the application (e.g., login, shopping cart, database writes) because those requests **still need to hit the** origin server. It reduces load **but** **DOESN'T** provide the **elastic compute capacity** **needed** for the application logic **during a** burst. It's a **good *complementary*** step but **NOT** a standalone solution for the core problem of handling sudden, unpredictable increases in *application processing* traffic.

In summary, **Option D** directly addresses the need for **elasticity in the web tier** using AWS's Auto Scaling **and** ELB, while allowing the **existing on-premises database to remain**, thus providing **a rapid** and **effective solution** for **handling traffic bursts** without a lengthy **and** complex full migration.

Q 103/218 You manually launch a **NAT(**Network Address Translation**) AMI** in a **public subnet. The** network is properly configured**. Security groups** and **Network Access Control Lists** are property configured**. Instances in a private subnet** can access the **NAT.** The **NAT** can access the **Internet. However, private instances CANNOT access the Internet. What additional step is required to allow access from the private instances?**

**A.** Disable Source/Destination Check on the NAT instance.

**B.** Enable Source/Destination Check on the private Instances.

**C.** Disable Source/Destination Check on the private instances.

**D.** Enable Source/Destination Check on the NAT instance.

**~~InCorrect Answer: C~~** Explanation When you manually set up a NAT instance (as opposed to using a managed NAT Gateway), it needs to perform Network Address Translation (NAT) for traffic originating from other instances. **This means** the NAT instance **will be sending** and **receiving** traffic where the ***source or destination IP*** address is **NOT its own**.

By default, every EC2 instance has a **Source/Destination Check** attribute enabled. This security feature ensures that the instance is only sending and receiving traffic for its own assigned IP addresses. If an instance attempts to send traffic on behalf of another IP address (like a NAT instance would), this check will prevent it.

Since your NAT instance needs to act as a go-between for your private instances and the internet, it must be able to send and receive traffic on behalf of the private instances. Disabling the Source/Destination Check allows the NAT instance to forward traffic for other IP addresses, enabling the private instances to reach the internet.

**Summary of Common NAT Instance Configuration Steps:**

To ensure a manually configured NAT instance works correctly**, THESE ARE** the typical requirements:

1. **NAT Instance** in a **Public Subnet:** The NAT instance MUST reside in a public subnet, meaning **its route table** has a **default route** to an **Internet Gateway**.
2. **Elastic IP (EIP)** for **NAT Instance:** The NAT instance needs a public IP address, typically an Elastic IP, for internet connectivity.
3. **Security Group** for **NAT Instance:**
   * **Inbound:** Allow HTTP (port 80) and HTTPS (port 443) from the private subnet's CIDR range. Also, allow SSH (port 22) from your administration IPs if you need to manage the NAT instance directly.
   * **Outbound:** Allow all traffic (0.0.0.0/0) on desired ports (e.g., 80, 443) to the internet.
4. **Network ACLs (NACLs):** Ensure NACLs on both the public and private subnets allow the necessary inbound and outbound traffic for NAT to function (e.g., ephemeral ports for return traffic).
5. **Route Table** for **Private Subnet:** The **route table** associated with the **private** subnet MUST have a **default** route (0.0.0.0/0) pointing to the **Network Interface ID (ENI)** of the NAT instance as its target.
6. **Disable Source/Destination Check** on **NAT instance:** This is **the crucial step** often missed, as explained above.

Given that the network is properly configured, security groups and NACLs are properly configured, and the NAT can access the internet, the missing piece for private instances to gain internet access through the NAT instance is almost certainly the **Source/Destination Check**.

**Given the Four options, it explained further as:**The best option to allow private instances to access the Internet through your manually launched NAT AMI, given the described setup, is **A. Disable Source/Destination Check on the NAT instance.**

**Explanation of Why option A is Correct:**

**A. Disable Source/Destination Check on the NAT instance.**

This is the critical step when setting up a NAT instance. By default, every EC2 instance performs a "source/destination check." This means the instance must be either the source or the destination of any network traffic it sends or receives.

However, a NAT instance acts as a router/proxy. It receives traffic *from* private instances (where the source IP is the private instance's IP, not the NAT instance's) and forwards that traffic *to* the internet (where the source IP is then translated to the NAT instance's public IP). When the response comes back from the internet, it's destined for the NAT instance's public IP, but the NAT instance then needs to forward it *to* the private instance.

If the Source/Destination Check is enabled on the NAT instance, it will drop any packet where its own IP is not the source or destination. Disabling this check allows the NAT instance to properly forward traffic on behalf of other instances.

**Why the Other Options Are Incorrect:**

**B. Enable Source/Destination Check on the private instances.**

* **Source/Destination** Check is **enabled b**y **defaul**t on **all EC2** instances, including your private instances. So, "enabling" it wouldn't be an additional step or change anything.
* More importantly, this setting on private instances has no bearing on their ability to send traffic through a NAT instance. Their job is simply to send traffic to the NAT instance **as their gateway**.

**C. Disable Source/Destination Check** on the **private instances.**

* This is **unnecessar**y and can introduce security risks. **Private** instances are **typically not** acting as **routers** or proxies, so there's no need for them to send or receive traffic on behalf of other IPs. **Disabling this check** on them **WOULDN'T** solve the NAT issue and could potentially allow them to be used for unintended routing if compromised.

**D. Enable Source/Destination Check on the NAT instance.**

* **This is the default state, and it's precisely why your private instances *cannot* access the Internet. The problem statement implies this is the current (incorrect) configuration. You need to disable it, not enable it.**

**In summary, for a manually configured NAT instance to correctly route traffic for other instances, you must explicitly disable its Source/Destination Check.**

Q 104/218 You've been hired to enhance the overall security posture for a very large e-commerce site They have a well architected multi-tier application running in a VPC that uses ELBs in front of both the web and the app tier with static assets served directly from S3 They are using a combination of RDS **and** DynamoDB for their dynamic data and then archiving **nightly** into S3 for further processing with **EMR** They are concerned because they found **questionable log** entries and **suspect** someone is attempting to gain **unauthorized** access.  
Which approach provides a **COST EFFECTIVE SCALABLE MITIGATION** to this kind of attack?

**A.** Add a **WAF tier** by creating a **new ELB** and an **AutoScalmg** group of **EC2 Instances** running a host-based WAF. They would **redirect Route 53** to resolve to the **new WAF tier ELB**. The WAF tier would ***then pass the traffic*** to the current web tier The web tier Security Groups would be updated to **ONLY ALLOW TRAFFIC** from the WAF tier Security Group

**B.** Add previously identified hostile source IPs as an explicit INBOUND DENY NACL to the web tier subnet.

**C.** Remove all but TLS 1 2 from the web tier ELB and enable Advanced Protocol Filtering This will enable the ELB itself to perform WAF functionality.

**D.** Recommend mat they lease space at a DirectConnect partner location and establish a 1G DirectConnect connection to their VPC. They would then establish Internet connectivity into their space, filter the traffic in hardware Web Application Firewall (WAF). And then pass the traffic through the DirectConnect connection into their application running in their VPC.

**Correct Answer: A**

Q 105/218 An AWS Lambda function requires access to an Amazon RDS for SQL Server instance. It is **against** company policy to store passwords in Lambda functions. How can a Solution Architect enable the Lambda function to retrieve the database password **WITHOUT** violating company policy?

**A.** Store a one-way hash of the password in the Lambda function

**B.** Have the Lambda function use the AWS Systems Manager Parameter Store

**C.** Add an IAM policy for IAM database access to the Lambda execution role

**D.** Connect to the Amazon RDS for SQL Server instance by using a role assigned to the Lambda function

**Correct Answer: A**

Q 106/218 Which of the following approaches provides the **LOWEST COST** for **Amazon Elastic Block Store snapshots** while giving you the **ability to fully restore data?**

**A.** Maintain a single snapshot the latest snapshot is both Incremental and complete.

**B.** Maintain a volume snapshot; subsequent snapshots will overwrite one another

**C.** Maintain two snapshots: the original snapshot and the latest incremental snapshot.

**D.** Maintain the most current snapshot, archive the original and incremental to Amazon Glacier.

**~~InCorrect Answer: C~~** Explanation**: The approach/option that provides the lowest cost for Amazon Elastic Block Store (EBS) snapshots** while retaining **the ability to fully restore data is option A. Maintain a single snapshot: the latest snapshot is both incremental and complete.**

**Explanation:** To understand why, it's crucial to remember how EBS snapshots work:

* **Incremental Nature: EBS snapshots are incremental. This means that only the blocks of data that have changed since the *previous* snapshot are stored. However, to restore a volume from *any* snapshot, AWS effectively reconstructs the full volume by combining the data from the initial full snapshot with all subsequent incremental snapshots up to the point of the chosen snapshot. AWS charges you only for the changed blocks.**

**Let's evaluate each option:**

**A. Maintain a single snapshot: the latest snapshot is both Incremental and complete.**

* This option refers to the inherent behavior of EBS snapshots. Even if you only retain the "latest" snapshot in your management view, AWS still stores all the necessary underlying incremental blocks from prior snapshots that are needed to reconstruct that "latest" complete state. Because you're only keeping what's essential to restore the *most recent* full state, you're only paying for the data that has changed between your last retained snapshot and the current one, plus the initial full snapshot's data. You pay for the unique blocks across *all* snapshots you keep. If you only keep one, you're paying for all the unique blocks that comprise that one point in time. This is the most cost-effective way to have *one point in time* restore.
* **Cost Implications:** You pay for the total amount of unique data blocks stored across all your snapshots. If you maintain only one logical snapshot (which implicitly includes all necessary previous unique blocks for its restoration), **you're paying** for the absolute **minimum unique data** required **for that** specific point in time.

**B. Maintain a volume snapshot; subsequent snapshots will overwrite one another.**

* EBS snapshots do not overwrite one another. Each snapshot creates a new point-in-time reference. AWS manages the underlying incremental blocks. If snapshots could overwrite each other, you would lose previous recovery points, which is contrary to the purpose of snapshots for backup and recovery. This statement describes a non-existent behavior.

**C. Maintain two snapshots: the original snapshot and the latest incremental snapshot.**

* This approach would involve keeping at least two logical points in time. You would pay for all the unique blocks that constitute the "original" snapshot, *plus* any unique blocks that constitute the "latest incremental" snapshot relative to the previous one. While this provides two recovery points, it will inherently cost more than maintaining just one, as you are storing more unique data.

**D. Maintain the most current snapshot, archive the original and incremental to Amazon Glacier.**

* EBS snapshots are *already* stored in Amazon S3, which is the underlying storage for EBS snapshots. There's no direct "archive to Amazon Glacier" action for an active EBS snapshot. While you can move an AMI-backed snapshot to a colder tier (like Glacier Flexible Retrieval or Glacier Deep Archive) using AWS Backup lifecycle policies, standard EBS snapshots are managed by the EBS service. Even if such a manual "archive" were possible, actively archiving and then restoring from Glacier would introduce additional costs (retrieval fees, potential delays) and management overhead, making it more expensive and complex than simply letting EBS manage the incremental storage for active snapshots.

Therefore, the lowest cost while still providing the ability to fully restore data (from that specific point in time) comes from understanding that EBS snapshots are already incremental and you only pay for the unique data blocks across all your retained snapshots. If you only need the *latest* recovery point, ensuring you keep only that single, most recent snapshot (which implicitly includes all necessary older blocks for its reconstruction) is the most cost-effective approach.

**SNAPSHOTS : Amazon Elastic Block Store (EBS) snapshots** do give you the ability to **fully restore data.1** They are a fundamental part of AWS's **data backup and recovery strategy** for **EC2 instances.**

**What is an EBS Snapshot?**

**An EBS snapshot is a point-in-time backup of your EBS volume.2 When you create a snapshot, AWS takes a copy of your volume's data at that precise moment and stores it in Amazon S3.3 Although stored in S3, you interact with snapshots through the EBS console or API; you don't directly access them in S3 buckets.4**

**Key Characteristics:**

* **Incremental:** After the first full snapshot, subsequent snapshots only save the blocks of data that have changed since the previous snapshot.5 This makes them highly efficient in terms of storage cost and creation time. However, when you restore from any snapshot, AWS reconstructs the full volume from the initial snapshot and all subsequent incremental changes.
* **Durable and Highly Available:** Snapshots are stored durably in Amazon S3, which is designed for 99.999999999% (11 nines) durability.6
* **Crash-Consistent (by default):** EBS snapshots are generally crash-consistent, meaning they capture the state of the volume as if the instance crashed at the moment of the snapshot.7 For applications like databases that require higher consistency (application-consistent), you might need to pause I/O or use application-specific backup methods (like VSS for Windows or xfs\_freeze for Linux) before taking the snapshot, or use AWS Backup which can orchestrate application-consistent backups.

**How to Fully Restore Data from an EBS Snapshot.** Restoring data from an EBS snapshot primarily involves creating a new EBS volume from that snapshot.8 You **CANNOT** restore a snapshot **onto an** **existing** EBS volume.

**Here's the general process:**

1. **Select the Snapshot: In the AWS Management Console (EC2 service -> Elastic Block Store -> Snapshots), select the specific snapshot you want to restore from.9**
2. **Create a New Volume from the Snapshot:**
   * Choose the option to "Create Volume from Snapshot."
   * You'll specify the volume type, size, and the Availability Zone (AZ) where the new volume should be created.10 *Important: The new volume must be created in the same AZ as the EC2 instance to which you intend to attach it.*11
   * The new volume will be an exact replica of the original volume at the point in time the snapshot was taken. The data is "lazily loaded," meaning it's streamed from S3 as it's accessed.12 For immediate full performance, you might choose to pre-initialize the volume or use features like Fast Snapshot Restore (FSR) if enabled on the snapshot.13
3. **Attach the New Volume to an EC2 Instance:**
   * Once the new volume is in the available state, you can attach it to an existing EC2 instance.
   * If you're **replacing a root** volume, you'll typically **need to stop** the EC2 instance first**, detach the old** root volume, **attach the new** volume **with the** correct device name (e.g., /dev/xvda for Linux root volumes), **and then** start the instance.
   * If it's a **data volume**, you can often **attach** it to a running instance **without downtime**.
4. **Mount and Access Data (within the OS):**

* After attaching the volume, you'll need to connect to the EC2 instance and mount the newly attached volume within the operating system (e.g., using mount command on Linux or Disk Management on Windows).14
* Once mounted, your application or users can access the data, which is now fully restored to the state it was in at the time the snapshot was taken.

**Use Cases for Full Restoration:**

* **Disaster Recovery**: Recovering from accidental data deletion, corruption, or an Availability Zone outage.15
* **Migration:** Moving data across different Availability Zones or AWS regions (by copying the snapshot to the target region first).16
* **Development/Testing**: Creating new environments with production data for testing, development, or analysis without impacting the live production volume.17

**In summary, EBS snapshots are a robust and reliable way to ensure you can fully recover your data from a specific point in time.18**

Q 107/218 A company is preparing to give AWS Management Console access to developers.  
Company policy mandates identity federation and role based access control. Roles are currently assigned using groups in the corporate Active Directory. What combination of the following will give developers access to the AWS console? Choose 2 answers

**A.** AWS identity and Access Management roles

**B.** AWS identity and Access Management groups

**C.** AWS identity and Access Management users

**D.** AWS Directory Service AD connector

**E.** AWS Directory Service Simple AD

**Correct Answer: A,D**

Q 108/218 **A weather forecasting company needs to process hundreds of gigabytes of data with sub-mill (second latency.  
The company has a high performance computing (HPC) environment in its data center and wants to expand its forecasting capabilities A solutions architect must identify a highly available cloud storage solution that can handle large amounts of sustained throughput Files that are stored in the solution should be accessible to thousands of compute instances that will simultaneously access and process the entire dataset What should the solutions architect do to meet these requirements?**

**A.** Use Amazon FSx for Lustre scratch file systems.

**B.** Use Amazon Elastic File System (Amazon EFS) with Bursting Throughput mode

**C.** Use Amazon FSx for Lustre persistent file systems

**D.** Use Amazon Elastic File System (Amazon EFS) with Provisioned Throughput mode

**Correct Answer: C**

Q 109/218 **An Elastic IP address (EIP) is a static IP address designed for dynamic cloud computing. With an EIP, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Your EIP is associated with your AWS account, not a particular EC2 instance, and it remains associated with your account until you choose to explicitly release it. By default how many EIPs is each AWS account limited to on a per region basis?**

**A.** 1

**B.** 5

**C.** Unlimited

**D.** 10

**Correct Answer: B** By default, all AWS accounts are limited to 5 Elastic IP addresses per region for each AWS account, because public (IPv4) Internet addresses are a scarce public resource. AWS strongly encourages you to use an EIP primarily for load balancing use cases, and use DNS hostnames for all other inter-node communication.  
If you feel your architecture warrants additional EIPs, you would need to complete the Amazon EC2  
Elastic IP Address Request Form and give reasons as to your need for additional addresses.  
Reference: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#using-instance-ad dressing-limit

Q 110/218 **You have been asked to design a fault-tolerant and scalable web application across three availability zones. The presentation logic will reside on web servers behind an ELB classic load balancer, and the application logic will reside on a set of app servers behind a second load balancer. How should you use auto scaling groups?**

**A.** Deploy one auto scaling group that includes al the web and app servers across all availability zones

**B.** Deploy two Auto Scaling groups: one for the web servers in all Availability zones and one for the app servers in all Availability zones

**C.** Deploy six auto scaling groups: a web server group in each Availability zone and an app server group in each availability zone

**D.** Deploy three auto scaling groups: one for each Availability zone that includes both web and app servers

**Correct Answer: D**

Q 111/218 **True or false: A VPC contains multiple subnets, where each subnet can span multiple Availability Zones.**

**A.** This is true only if requested during the set-up of VPC.

**B.** This is true.

**C.** This is false.

**D.** This is true only for US regions.

**Correct Answer: C**

A VPC can span several Availability Zones. In contrast, a subnet must reside within a single Availability Zone.  
Reference: https://aws.amazon.com/vpc/faqs/

Q 112/218 **A web application is deployed in the AWS Cloud. It consists of a two-tier architecture that includes a web layer and a database layer. The web server is vulnerable to cross-site scripting (XSS) attacks.  
What should a solutions architect do to remediate the vulnerability?**

**A.** Create a Classic Load Balancer. Put the web layer behind the load balancer and enable AWS WAF.

**B.** Create a Network Load Balancer. Put the web layer behind the load balancer and enable AWS WAF.

**C.** Create an Application Load Balancer. Put the web layer behind the load balancer and enable AWS WAF.

**D.** Create an Application Load Balancer. Put the web layer behind the load balancer and use AWS Shield Standard.

**Correct Answer: C**

Working with cross-site scripting match conditions  
Attackers sometimes insert scripts into web requests in an effort to exploit vulnerabilities in web applications.  
You can create one or more cross-site scripting match conditions to identify the parts of web requests, such as the URI or the query string, that you want AWS WAF Classic to inspect for possible malicious scripts. Later in the process, when you create a web ACL, you specify whether to allow or block requests that appear to contain malicious scripts.  
Web Application Firewall  
You can now use AWS WAF to protect your web applications on your Application Load Balancers. AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.  
Reference: https://docs.aws.amazon.com/waf/latest/developerguide/classic-web-acl-xss-conditions.html  
https://aws.amazon.com/elasticloadbalancing/features/

Q  113/218 **A company copies 200 TB of data from a recent ocean survey onto AWS Snowball Edge Storage Optimized devices The company has a high performance computing (HPC) cluster that is hosted on AWS to look for oil and gas deposits A solutions architect must provide the cluster with consistent sub-millisecond latency and high-throughput access to the data on the Snowball Edge Storage Optimized devices The company is sending the devices back to AWS Which solution will meet these requirements'?**

**A.** Create an Amazon S3 bucket and an Amazon Elastic File System (Amazon EFS) file system Import the data into the S3 bucket Copy the data from the S3 bucket to the EFS file system Access the EFS file system from the HPC cluster instances

**B.** Create an Amazon FSx for Lustre file system Import the data directly into the FSx for Lustre file system Access the FSx for Lustre file system from the HPC cluster instances

**C.** Create an Amazon S3 bucket Import the data into the S3 bucket. Configure an AWS Storage Gateway file gateway to use the S3 bucket Access the file gateway from the HPC cluster instances

**D.** Create an Amazon S3 bucket Import the data into the S3 bucket Configure an Amazon FSx for Lustre file system and integrate it with the S3 bucket Access the FSx for Lustre file system from the HPC cluster instances

**Correct Answer: C**

Q 114/218 **An organization hosts an app on EC2 instances which multiple developers need access to in order to perform updates.  
The organization plans to implement some security best practices related to instance access.  
Which one of the following recommendations will NOT help improve its security in this way?**

**A.** Disable the password based login for all the users. All the users should use their own keys to connect with the instance securely.

**B.** Create an IAM policy allowing only IAM users to connect to the EC2 instances with their own SSH key.

**C.** Create a procedure to revoke the access rights of the individual user when they are not required to connect to EC2 instance anymore for the purpose of application configuration.

**D.** Apply the latest patch of OS and always keep it updated.

**Correct Answer: B** Since AWS is a public cloud any application hosted on EC2 is prone to hacker attacks. It becomes extremely important for a user to setup a proper security mechanism on the EC2 instances. A few of the security measures are listed below:  
\* Always keep the OS updated with the latest patch  
\* Always create separate users with in OS if they need to connect with the EC2 instances, create their keys and disable their password  
\* Create a procedure using which the admin can revoke the access of the user when the business work on the EC2 instance is completed. . Lock down unnecessary ports  
\* Audit any proprietary applications that the user may be running on the EC2 instance. Provide temporary escalated privileges, such as sudo for users who need to perform occasional

Privileged tasks IAM **is useful** when users are required to work with AWS resources and actions, such as launching an instance. **It is not** useful **in this** case because it does not manage who can connect via RDP or SSH with an instance.  
http://aws.amazon.com/articles/1233/

Q 115/218 **How does AWS Data Pipeline execute activities on on-premise resources or AWS resources that you manage?**

**A.** By supplying a Task Runner package that can be installed on your on-premise hosts

**B.** None of these

**C.** By supplying a Task Runner file that the resources can access for execution

**D.** By supplying a Task Runner json script that can be installed on your on-premise hosts

**Correct Answer: A** To enable running activities using on-premise resources, AWS Data Pipeline does the following: It supply a Task Runner package that can be installed on your on-premise hosts. This package continuously polls the AWS Data Pipeline service for work to perform. When it's time to run a particular activity on your on-premise resources, it will issue the appropriate command to the Task Runner.  
https://aws.amazon.com/datapipeline/faqs/

Q 116/218 **You are tasked with setting up a Linux bastion host for access to Amazon EC2 instances running in your VPC. Only clients connecting from the corporate external public IP address 72.34.51.100 should have SSH access to the host. Which option will meet the customer requirement?**

**A.** Security Group Inbound Rule: Protocol - UDP, Port Range - 22, Source 72.34.51.100/32

**B.** Security Group Inbound Rule: Protocol - TCP. Port Range - 22, Source 72.34.51.100/32

**C.** Network ACL Inbound Rule: Protocol - TCP, Port Range-22, Source 72.34.51.100/0

**D.** Network ACL Inbound Rule: Protocol - UDP, Port Range - 22, Source 72.34.51.100/32

**Correct Answer: B**

Q 117/218

**You are playing around with setting up stacks using JSON templates in CloudFormation to try and understand them a little better. You have set up about 5 or 6 but now start to wonder if you are being charged for these stacks.  
What is AWS's billing policy regarding stack resources?**

**A.** You are not charged for the stack resources if they are not taking any traffic.

**B.** You are charged for the stack resources for the time they were operating (but not if you deleted the stack within 30 minutes)

**C.** You are charged for the stack resources for the time they were operating (but not if you deleted the stack within 60 minutes)

**D.** You are charged for the stack resources for the time they were operating (even if you deleted the stack right away)

**Correct Answer: D**

Explanation/Reference:  
Explanation:  
A stack is a collection of AWS resources that you can manage as a single unit. In other words, you can create, update, or delete a collection of resources by creating, updating, or deleting stacks. All the resources in a stack are defined by the stack's AWS CloudFormation template. A stack, for instance, can include all the resources required to run a web application, such as a web server, a database, and networking rules. If you no longer require that web application, you can simply delete the stack, and all of its related resources are deleted. You are charged for the stack resources for the time they were operating (even if you deleted the stack right away).  
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacks.html>

Q 118/218

**A solutions architect needs to migrate 50 TB of NFS data to Amazon S3. The files are on several NFS file servers on corporate network. These are dense file systems containing tens of millions of small files. The system operators have configured the file interface on an AWS Snowball Edge device and are using a shell script to copy data.  
Developers report that copying the data to the Snowball Edge device is very slow. The solutions architect suspects this may be related to the overhead of encrypting all the small files and transporting them over the network.  
Which changes can be made to speed up the data transfer?**

**A.** Cluster two Snowball Edge devices together to increase the throughput of the devices.

**B.** Connect directly to the USB interface on the Snowball Edge device and copy the files locally.

**C.** Increase the number of parallel copy jobs to increase the throughput of the Snowball Edge device.

**D.** Change the solution to use the S3 Adapter instead of the file interface on the Snowball Edge device.

**Correct Answer: D**

Q 119/218

**In Route 53, what does a Hosted Zone refer to?**

**A.** A hosted zone is a collection of geographical load balancing rules for Route 53.

**B.** A hosted zone is a collection of resource record sets hosted by Route 53.

**C.** A hosted zone is a selection of specific resource record sets hosted by CloudFront for distribution to Route 53.

**D.** A hosted zone is the Edge Location that hosts the Route 53 records for a user.

**Correct Answer: B**

A Hosted Zone refers to a selection of resource record sets hosted by Route 53.  
Reference: http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/AboutHostedZones.html

Q  120/218

**A company needs guaranteed Amazon EC2 capacity in three specific Availability Zones in a specific AWS Region for an upcoming event that will last 1 week. What should the company do to guarantee the EC2 capacity?**

**A.** Create an On-Demand Capacity Reservation that specifies the Region and three Availability Zones needed.

**B.** Purchase Reserved Instances that specify the Region needed.

**C.** Purchase Reserved Instances that specify the Region and three Availability Zones needed.

**D.** Create an On-Demand Capacity Reservation that specifies the Region needed.

**Correct Answer: A**

Q 121/218

**Your supervisor has asked you to build a simple file synchronization service for your department. He doesn't want to spend too much money and he wants to be notified of any changes to files by email. What do you think would be the best Amazon service to use for the email solution?**

**A.** Amazon SES

**B.** Amazon CloudSearch

**C.** Amazon SWF

**D.** Amazon AppStream

**Correct Answer: A**

File change notifications can be sent via email to users following the resource with Amazon Simple Email  
Service (Amazon SES), an easy-to-use, cost-effective email solution.  
Reference: http://media.amazonwebservices.com/architecturecenter/AWS\_ac\_ra\_filesync\_08.pdf

Q 122/218

**A media company asked a Solutions Architect to design a highly available storage solution to serve as a centralized document store for their Amazon EC2 instances. The storage solution needs to be POSIX-compliant, scale dynamically, and be able to serve up to  
100 concurrent EC2 instances.  
Which solution meets these requirements?**

**A.** Create an Amazon S3 bucket and store all of the documents in this bucket.

**B.** Create an Amazon EBS volume and allow multiple users to mount that volume to their EC2 instance(s).

**C.** Use Amazon Glacier to store all of the documents.

**D.** Create an Amazon Elastic File System (Amazon EFS) to store and share the documents.

**Correct Answer: D**

Explanation  
https://docs.aws.amazon.com/efs/latest/ug/creating-using.html  
Creating Resources for Amazon EFS Amazon EFS provides elastic, shared file storage that is POSIX-compliant. The file system you create supports concurrent read and write access from multiple Amazon EC2 instances and is accessible from all of the Availability Zones in the AWS Region where it is created.

Q  123/218

**A company wants to run a hybrid workload for data processing. The data needs to be accessed by on-premises applications for local data processing using an NFS protocol, and must also be accessible from the AWS Cloud for further analytics and batch processing.  
Which solution will meet these requirements?**

**A.** Use an AWS storage Gateway tape gateway to copy the backup of the local data to AWS, then perform analytics on this data in the AWS cloud.

**B.** Use an AWS Storage Gateway volume gateway in a cached volume configuration to back up all the local storage in the AWS cloud, then perform analytics on this data in the cloud.

**C.** Use an AWS Storage Gateway volume gateway in a stored volume configuration to regularly take snapshots of the local data, then copy the data to AWS.

**D.** Use an AWS Storage Gateway file gateway to provide file storage to AWS, then perform analytics on this data in the AWS Cloud.

**Correct Answer: C**

Q 124/218

**After an Amazon VPC instance is launched, can I change the VPC security groups it belongs to?**

**A.** No. You cannot.

**B.** Yes. You can.

**C.** Only if the tag "VPC Change Group" is true

**D.** Only if the tag "VPC\_Change\_Group" is true

**Correct Answer: B**

Q 125/218

**A company collects temperature, humidity, and atmospheric pressure data in cities across multiple continents.  
The average volume of data collected per site each day is 500 GB. Each site has a high-speed internet connection. The company's weather forecasting applications are based in a single Region and analyze the data daily.  
What is the FASTEST way to aggregate data for all of these global sites?**

**A.** Enable Amazon S3 Transfer Acceleration on the destination bucket. Use multipart uploads to directly upload site data to the destination bucket.

**B.** Upload site data to an Amazon S3 bucket in the closest AWS Region. Use S3 cross-Region replication to copy objects to the destination bucket.

**C.** Upload the data to an Amazon EC2 instance in the closes Region. Store the data in an Amazon EBS volume. One a day take an EBS snapshot and copy it to the centralize Region. Restore the EBS volume in the centralized Region and run an analysis on the data daily.

**D.** Upload site data to an Amazon S3 bucket in the closest AWS Region. Use S3 cross-Region replication to copy objects to the destination bucket.

**Correct Answer: C**

Q 126/218

**A company is migrating an application to AWS. It wants to use fully managed services as much as possible during the migration. The company needs to store large, important documents within the application with the following requirements:  
\* The data must be highly durable and available.  
\* The data must always be encrypted at rest and in transit.  
\* The encryption key must be managed by the company and rotated periodically.  
Which of the following solutions should the Solutions Architect recommend?**

**A.** Deploy the storage gateway to AWS in file gateway mode. Use Amazon EBS volume encryption using an AWS KMS key to encrypt the storage gateway volumes.

**B.** Use Amazon DynamoDB with SSL to connect to DynamoDB. Use an AWS KMS key to encrypt DynamoDB objects at rest.

**C.** Use Amazon S3 with a bucket policy to enforce HTTPS for connections to the bucket and to enforce server-side encryption and AWS KMS for object encryption.

**D.** Deploy instances with Amazon EBS volumes attached to store this data. Use EBS volume encryption using an AWS KMS key to encrypt the data.

**Correct Answer: A**

Q  127/218

**Which of the following statements is NOT correct when working with your AWS Direct Connect connection after it is set up completely?**

**A.** You can manage your AWS Direct Connect connections and view the connection details.

**B.** You can delete a connection as long as there are no virtual interfaces attached to it.

**C.** You cannot view the current connection ID and verify if it matches the connection ID on the Letter of Authorization (LOA).

**D.** You can accept a host connection by purchasing a hosted connection from the partner (APN).

**Correct Answer: C**

You can manage your AWS Direct Connect connections and view connection details, accept hosted connections, and delete connections. You can view the current status of your connection. You can also view your connection ID, which looks similar to this example dxcon-xxxx, and verify that it matches the connection ID on the Letter of Authorization (LOA) that you received from Amazon.  
http://docs.aws.amazon.com/directconnect/latest/UserGuide/viewdetails.html

Q 128/218

**Fill in the blanks: \_\_ is a durable, block-level storage volume that you can attach to a single, running Amazon EC2 instance.**

**A.** None of these

**B.** Amazon EBS

**C.** All of these

**D.** Amazon 53

**Correct Answer: B**

Q 129/218

**A company is deploying a two-tier, highly available web application to AWS. Which service provides durable storage for static content while utilizing lower Overall CPU resources for the web tier?**

**A.** Amazon 53

**B.** Amazon EBS volume

**C.** Amazon RD5 instance

**D.** Amazon EC2 instance store

**Correct Answer: A**

Q 130/218

**A company has a Microsoft NET application that runs on an on-premises Windows Server. The application stores data by using an Oracle Database Standard Edition server. The company is planning a migration to AWS and wants to minimize development changes while moving the application. The AWS application environment should be highly available.  
Which combination of actions should the company take to meet these requirements? (**Select TWO.**)**

**A.** Rehost the application in AWS Elastic Beanstalk with the .NET platform in a Multi-AZ deployment.

**B.** Replatform the application to run on Amazon EC2 with the Amazon Linus Amazon Machine Image (AMI).

**C.** Use AWS Database Migration Service (AWS DMS) to migrate from the Oracle database to Oracle on Amazon RDS in a Multi-AZ deployment.

**D.** Refactor the application as serverless with AWS Lambda functions running NET Core.

**E.** Use AWS Database Migration Service (AWS DMS) to migrate from the Oracle database to Amazon DynamoDB in a Multi-AZ deployment.

**Correct Answer: D,E**

Q  131/218

**A workload in an Amazon VPC consists of a single web server launched from a custom AMI.  
Session state is stored in a database.  
How should the Solutions Architect modify this workload to be both highly available and scalable?**

**A.** Create a launch configuration with a desired capacity of two web servers across multiple Availability Zones. Create an Auto Scaling group with the AMI ID of the web server image. Use Amazon Route 53 latency-based routing to balance traffic across the Auto Scaling group.

**B.** Create a launch configuration with the AMI ID of the web server image. Create an Auto Scaling group using the newly-created launch configuration, and a desired capacity of two web servers across multiple Availability Zones. Use Amazon Route 53 weighted routing to balance traffic across the Auto Scaling group.

**C.** Create a launch configuration with the AMI ID of the web server image. Create an Auto Scaling group using the newly-created launch configuration, and a desired capacity of two web servers across multiple Availability Zones. Use an ALB to balance traffic across the Auto Scaling group.

**D.** Create a launch configuration with the AMI ID of the web server image. Create an Auto Scaling group using the newly-created launch configuration, and a desired capacity of two web servers across multiple regions. Use an Application Load Balancer (ALB) to balance traffic across the Auto Scaling group.

**Correct Answer: C**

Q 132/218

**Amazon RDS automated backups and DB Snapshots are currently supported for only the \_\_\_ \_ storage engine**

**A.** lnnoDB

**B.** MyISAM

**Correct Answer: A**