questoes erradas-1

Notebook:	solutions profissional_ef8e1c0f-5d3c-42ab-ab04-9217e53e12cd		
Created:	04/11/2025 21:39	**Updated:**	10/12/2025 09:06
Author:	erikachen19@gmail.com		
URL:	https://portal.tutorialsdojo.com/courses/aws-certified-solutions-architect-professional...		

To route domain traffic to an ELB load balancer,
use Amazon Route 53 to create an **alias record** -> load balancer.
An alias record is a Route 53 extension to DNS. It's similar to a CNAME record, but you can create an alias record both for the root domain, such as example.com, and for subdomains, such as www.example.com. (You can create CNAME records only for subdomains).
aplication load balancer. -> alias record
For EC2 instances, -> Type A Record (no Alias.)
For ELB, Cloudfront, and S3, Type A Alias Record
RDS, CNAME Record (no Alias.)

2. QUESTION

A company has a hybrid set up for its mobile application. The on-premises data center hosts a 3TB MySQL database server that handles the write-intensive requests from the application. The on-premises network is connected to the AWS VPC with a VPN. On AWS, the serverless application runs on AWS Lambda and API Gateway with an Amazon DynamoDB table used for saving user preferences. The application scales well as more users are using the mobile app. The user traffic is unpredictable but there is an average increase of about 20% each month. A few months into operation, the company noticed the exponential increase of costs for AWS Lambda. The Solutions Architect noticed that the Lambda execution time averages 4.5 minutes and most of that is wait time due to latency when calling the on-premises data MySQL server.

Which of the following solutions should the Solutions Architect implement to reduce the overall cost?

- 1.Provision an AWS Direct Connect connection from the on-premises data center to Amazon VPC instead of a VPN to significantly reduce the network latency to the MySQL server.
  2. Configure caching on the mobile application to reduce the overall AWS Lambda function calls.
  3. Gradually lower the timeout and memory properties of the Lambda functions without increasing the execution time.
  4. Add an Amazon Elasticache cluster in front of DynamoDB to cache the frequently accessed records.

- 1. Migrate the on-premises MySQL database server to Amazon RDS for MySQL. Enable Multi-AZ to ensure high availability.
  2. Configure API caching on Amazon API Gateway to reduce the overall number of invocations to the Lambda functions.
  3. Gradually lower the timeout and memory properties of the Lambda functions without increasing the execution time.
  4. Configure Auto Scaling on Amazon DynamoDB to automatically adjust the capacity based on user traffic.

- (selected) 1. Provision an AWS Direct Connect connection from the on-premises data center to Amazon VPC instead of a VPN to significantly reduce the network latency to the MySQL server.
  2. Create a CloudFront distribution with the API Gateway as the origin to cache the API responses and reduce the Lambda invocations.
  3. Convert the Lambda functions to run them on Amazon EC2 Reserved Instances. Use Auto Scaling on peak time with a combination of Spot instances to further reduce costs.
  4. Configure Auto Scaling on Amazon DynamoDB to automatically adjust the capacity with user traffic.

- 1. Migrate the on-premises MySQL database server to Amazon RDS for MySQL. Enable Multi-AZ to ensure high availability.
  2. Create a CloudFront distribution with the API Gateway as the origin to cache the API responses and reduce the Lambda invocations.
  3. Gradually lower the timeout and memory properties of the Lambda functions without increasing the execution time.
  4. Configure Auto Scaling on Amazon DynamoDB to automatically adjust the capacity with user traffic and enable DynamoDB Accelerator to cache frequently accessed records.

6. QUESTION

A retail company hosts its web application on an Auto Scaling group of Amazon EC2 instances deployed across multiple Availability Zones. The Auto Scaling group is configured to maintain a minimum EC2 cluster size and automatically replace unhealthy instances. The EC2 instances are behind an Application Load Balancer so that the load can be spread evenly on all instances. The application target group health check is configured with a fixed HTTP page that queries a dummy item on the database. The web application connects to a Multi-AZ Amazon RDS MySQL instance. A recent outage caused a major loss to the company's revenue. Upon investigation, it was found that the web server metrics are within the normal range but the database CPU usage is very high, causing the EC2 health checks to timeout. Failing the health checks, the Auto Scaling group continuously replaced the unhealthy instances thus causing the downtime.

Which of the following options should the Solution Architect implement to prevent this from happening again and allow the application to handle more traffic in the future? (Select TWO.)

- [ ] Create an Amazon CloudWatch alarm to monitor the Amazon RDS MySQL instance if it has a high-load or in impaired status. Set the alarm action to recover the RDS instance. This will automatically reboot the database to reset the queries.

- [x] Reduce the load on the database tier by creating an Amazon ElastiCache cluster to cache frequently requested database queries. Configure the application to use this cache when querying the RDS MySQL instance.

- [x] Change the target group health check to use a TCP check on the EC2 instances instead of a page that queries the database. Create an Amazon Route 53 health check for the database dummy item web page to ensure that the application works as expected. Set up an Amazon CloudWatch alarm to send a notification to Admins when the health check fails.

- [ ] Reduce the load on the database tier by creating multiple read replicas for the Amazon RDS MySQL Multi-AZ cluster. Configure the web application to use the single reader endpoint of RDS for all read operations.

- [ ] Change the target group health check to a simple HTML page instead of a page that queries the database. Create an Amazon Route 53 health check for the database dummy item web page to ensure that the application works as expected. Set up an Amazon CloudWatch alarm to send a notification to Admins when the health check fails.

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution. At the same time, it helps remove the complexity associated with deploying and managing a distributed cache environment.

ElastiCache for Memcached has multiple features to enhance reliability for critical production deployments:

– Automatic detection and recovery from cache node failures.

– Automatic discovery of nodes within a cluster enabled for automatic discovery so that no changes need to be made to your application when you add or remove nodes.

– Flexible Availability Zone placement of nodes and clusters.

– Integration with other AWS services such as Amazon EC2, Amazon CloudWatch, AWS CloudTrail, and Amazon SNS to provide a secure, high-performance, managed in-memory caching solution.

The option that says: **Change the target group health check to a simple HTML page instead of a page that queries the database. Create an Amazon Route 53 health check for the database dummy item web page to ensure that the application works as expected. Set up an Amazon CloudWatch alarm to send a notification to Admins when the health check fails** is correct. Changing the target group health check to a simple HTML page will reduce the queries to the database tier. The Route 53 health check can act as the "external" check on a specific page that queries the database to ensure that the application is working as expected. The Route 53 health check has an overall lower request count compared to using the target group health check.

The option that says: **Reduce the load on the database tier by creating an Amazon ElastiCache cluster to cache frequently requested database queries. Configure the application to use this cache when querying the RDS MySQL instance** is correct. Since this is a retail web application, most of the queries will be read-intensive as customers are searching for products. ElastiCache is effective at caching frequent requests, which overall improves the application response time and reduces database queries.

7. QUESTION

A multinational investment bank has a hybrid cloud architecture that uses a single 1 Gbps AWS Direct Connect connection to integrate their on-premises network to AWS Cloud. The bank has a total of 10 VPCs which are all connected to their on-premises data center via the same Direct Connect connection that you manage. Based on the recent IT audit, the existing network setup has a single point of failure which needs to be addressed immediately.

Which of the following is the MOST cost-effective solution that you should implement in order to improve the connection redundancy of your hybrid network?

- Establish another 1 Gbps AWS Direct Connect connection using a public Virtual Interface (VIF). Prepare a VPN tunnel that will terminate on the virtual private gateway (VGW) of the respective VPC using the public VIF. Handle the failover to the VPN connection through the use of BGP.
- Establish VPN tunnels from your on-premises data center to each of the 10 VPCs. Terminate each VPN tunnel connection at the virtual private gateway (VGW) of the respective VPC. Configure BGP for route management.
- Establish another 1 Gbps AWS Direct Connect connection with corresponding private Virtual Interfaces (VIFs) to connect all of the 10 VPCs individually. Set up a Border Gateway Protocol (BGP) peering session for all of the VIFs.
- Establish a new point-to-point Multiprotocol Label Switching (MPLS) connection to all of your 10 VPCs. Configure BGP to use this new connection with an active/passive routing.

10. QUESTION

A company has production, development, and test environments in its software development department, and each environment contains tens to hundreds of EC2 instances, along with other AWS services. Recently, Ubuntu released a series of security patches for a critical flaw that was detected in their OS. Although this is an urgent matter, there is no guarantee yet that these patches will be bug-free and production-ready hence, the company must immediately patch all of its affected Amazon EC2 instances in all the environments, except for the production environment. The EC2 instances in the production environment will only be patched after it has been verified that the patches work effectively. Each environment also has different baseline patch requirements that needed to be satisfied.

Using the AWS Systems Manager service, how should you perform this task with the least amount of effort?

- Schedule a maintenance period in AWS Systems Manager Maintenance Windows for each environment, where the period is after business hours so as not to affect daily operations. During the maintenance period, Systems Manager will execute a cron job that will install the required patches for each EC2 instance in each environment. After that, verify in Systems Manager Managed Instances that your environments are fully patched and compliant.
- Tag each instance based on its OS. Create a patch baseline in AWS Systems Manager Patch Manager for each environment. Categorize EC2 instances based on their tags using Patch Groups and then apply the patches specified in the corresponding patch baseline to each Patch Group. Afterward, verify that the patches have been installed correctly using Patch Compliance. Record the changes to patch and association compliance statuses using AWS Config.
- Tag each instance based on its environment and OS. Create various shell scripts for each environment that specifies which patch will serve as its baseline. Using AWS Systems Manager Run Command, place the EC2 instances into Target Groups and execute the script corresponding to each Target Group.
- Tag each instance based on its environment and OS. Create a patch baseline in AWS Systems Manager Patch Manager for each environment. Categorize EC2 instances based on their tags using Patch Groups and apply the patches specified in the corresponding patch baseline to each Patch Group.

11. QUESTION

A multi-national tech company has multiple VPCs assigned for each of its IT departments. VPC peering has been set up whenever intercommunication is needed between the VPCs. The solutions architect has been instructed to launch a new central database server that can be accessed by the other VPCs of the company using the `database.tutorialsdojo.com` domain name. This server should only be resolvable and accessible within the associated VPCs since only internal applications will be using the database.

Which of the following options should the solutions architect implement to meet the above requirements?

- Set up a public hosted zone with a domain name of tutorialsdojo.com and specify the VPCs that you want to associate with the hosted zone. Create a CNAME record with a value of database.tutorialsdojo.com which maps to the IP address of the EC2 instance of your database server. Modify the `enableDnsHostNames` attribute of your VPC to `false` and the `enableDnsSupport` attribute to `false`
- Set up a private hosted zone with a domain name of tutorialsdojo.com and specify the VPCs that you want to associate with the hosted zone. Create an A record with a value of database.tutorialsdojo.com which maps to the IP address of the EC2 instance of your database server. Modify the `enableDnsHostNames` attribute of your VPC to `true` and the `enableDnsSupport` attribute to `true`
- Set up a private hosted zone with a domain name of tutorialsdojo.com and specify the VPCs that you want to associate with the hosted zone. Create an A record with a value of database.tutorialsdojo.com which maps to the Elastic IP address of the EC2 instance of your database server. Modify the `enableDnsHostNames` attribute of your VPC to `true` and the `enableDnsSupport` attribute to `false`
- Set up a public hosted zone with a domain name of tutorialsdojo.com and specify the VPCs that you want to associate with the hosted zone. Create an A record with a value of database.tutorialsdojo.com which maps to the IP address of the EC2 instance of your database server. Modify the `enableDnsHostNames` attribute of your VPC to `true` and the `enableDnsSupport` attribute to `true`

– ***Public hosted zones*** contain records that specify how you want to route traffic on the internet.

– ***Private hosted zones*** contain records that specify how you want to route traffic in an Amazon VPC

you have to create a ***private*** hosted zone and not a public one, since the database server will only be accessed by the associated VPCs and not publicly over the Internet.

In addition, you have to create an A record for your database server and then set
both the **`enableDnsHostNames`** and **`enableDnsSupport`** attributes to `true`.

an **Elastic** IP address is a public IPv4 address, which is reachable from the Internet and hence,

your EC2 instance receives public DNS hostnames, and whether DNS resolution through the Amazon DNS server is supported.

DOMIN NAME `enableDnsHostnames` `TRUE` , `enableDnsSupport` `TRUE` Indicates whether the DNS resolution is supported for the VPC.

14. QUESTION

A global financial company is launching its new trading platform in AWS which allows people to buy and sell their bitcoin, ethereum, ripple, and other cryptocurrencies, as well as access to various financial reports. To meet the anti-money laundering and counter-terrorist financing (AML/CFT) measures compliance, all report files of the trading platform must not be accessible in certain countries which are listed in the Financial Action Task Force (FATF) list of non-cooperative countries or territories. You were given a task to ensure that the company complies with this requirement to avoid hefty monetary penalties.

In this scenario, what is the best way to satisfy this security requirement in AWS while still delivering content to users around the globe with lower latency?

- (●) Create a CloudFront distribution with Geo-Restriction enabled to block all of the blacklisted countries from accessing the trading platform.
- ( ) Use Route 53 with a Geolocation routing policy that blocks all traffic from the blacklisted countries.
- ( ) Use Route 53 with a Geoproximity routing policy that blocks all traffic from the blacklisted countries.
- ( ) Deploy the trading platform using Elastic Beanstalk and deny all incoming traffic from the IP addresses of the blacklisted countries in the Network Access Control List (ACL) of the VPC.

16. QUESTION

A leading financial company is planning to launch its Node.js application with an Amazon RDS MariaDB database to serve its clients worldwide. The application will run on both on-premises servers as well as Reserved EC2 instances. To comply with the company's strict security policy, the database credentials must be encrypted both at rest and in transit. These credentials will be used by the application servers to connect to the database. The Solutions Architect is tasked to manage all of the aspects of the application architecture and production deployment.

How should the Architect automate the deployment process of the application in the MOST secure manner?

- ( ) Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to all on-premises servers and EC2 instances. Use Elastic Beanstalk to host and manage the application on both on-premises servers and EC2 instances. Deploy the succeeding application revisions to AWS and on-premises servers using Elastic Beanstalk.
- ( ) Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to the EC2 instances. Create an IAM Service Role that will be associated with the on-premises servers. Deploy the application packages to the EC2 instances and on-premises servers using AWS CodeDeploy.
- (●) Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Attach this IAM policy to the instance profile for CodeDeploy-managed EC2 instances. Associate the same policy as well to the on-premises instances. Using AWS CodeDeploy, launch the application packages to the Amazon EC2 instances and on-premises servers.
- ( ) Upload the database credentials with key rotation in AWS Secrets Manager. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to all on-premises servers and EC2 instances. Use Elastic Beanstalk to host and manage the application on both on-premises servers and EC2 instances. Deploy the succeeding application revisions to AWS and on-premises servers using Elastic Beanstalk.

You can't deploy an application to your on-premises servers using Elastic Beanstalk. This is only applicable to your Amazon EC2 instances.

18. QUESTION

A startup is building a web app that lets users post photos of good deeds in their neighborhood with a 143-character caption/article. The developers decided to write the application in ReactJS, a popular javascript framework so that it would run on the broadest range of browsers, mobile phones, and tablets. The app should provide access to Amazon DynamoDB to store the caption. The initial prototype shows that there aren't large spikes in usage.

Which option provides the most cost-effective and scalable architecture for this application?

- Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or from any other popular social sites and use the `AssumeRoleWithWebIdentity` API of STS to generate temporary credentials. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in Amazon S3 and DynamoDB. Serve your web app out of an S3 bucket enabled as a website.
- Configure the ReactJS client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) on an EC2 instance. This will provide signed credentials to an IAM user allowing GET and PUT operations in the DynamoDB table and the S3 bucket. You serve your mobile application out of an S3 bucket enabled as a website.
- Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or any other popular social site. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in DynamoDB. Serve your web application from an NGINX server hosted on a fleet of EC2 instances, with a load balancer and auto-scaling. Add an IAM role to the EC2 instance to allow GET and PUT operations to DynamoDB tables. CARO
- Configure the ReactJS client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) to provide signed credentials to an IAM user. This will allow GET and PUT operations to DynamoDB. Serve your web application from an NGINX server hosted in a fleet of EC2 instances that are load-balanced and auto-scaled. Your EC2 instances are configured with an IAM role that allows GET and PUT operations in DynamoDB.

react site estatico Navegador React ,Vue, html css, Angular

site dimanico :

WordPress, Django. Flask, Node (Express, NestJS). Ruby on Rails, Java Spring, PHP Laravel

If you don't use Amazon Cognito, then you choose to write a custom code or app that interacts with a web IdP (Login with Amazon, Facebook, Google, or any other OIDC-compatible IdP) and then call the `AssumeRoleWithWebIdentity` API to trade the authentication token you get from those IdPs for AWS temporary security credentials. If you have already used this approach for existing apps, you can continue to use it. You can also deploy your app in the S3 bucket.

The option that says: **Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or any other popular social sites and use the `AssumeRoleWithWebIdentity` API of STS to generate temporary credentials. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in Amazon S3 and DynamoDB. Serve your web app out of an S3 bucket enabled as a website** is correct because it authenticates the application via a federated identity provider such as Google, Facebook, Amazon, or other social sites. It sets up proper permission for DynamoDB access and hosts the website in S3. Plus, it also uses STS and `AssumeRoleWithWebIdentity` API which provides a better authentication.

The option that says: **Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or from any other popular social site. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in DynamoDB. Serve your web application from an NGINX server hosted on a fleet of EC2 instances, with a load balancer and auto-scaling. Add an IAM role to the EC2 instance to allow GET and PUT operations to DynamoDB tables** is incorrect because it does not mention any security token service that generates temporary credentials. Furthermore, deploying EC2 instances in an auto-scaled environment, albeit scalable, is not as cost-effective as the S3 website.

The option that says: **Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or from any other popular social site. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in DynamoDB. Serve your web application from an NGINX server hosted on a fleet of EC2 instances, with a load balancer and auto-scaling. Add an IAM role to the EC2 instance to allow GET and PUT operations to DynamoDB tables** is incorrect because it does not mention any security token service that generates temporary credentials. Furthermore, deploying EC2 instances in an auto-scaled environment, albeit scalable, is not as cost-effective as the S3 website.

19. QUESTION

A company needs a deployment solution for its application that is hosted on the AWS cloud. The company has the following requirements for the application:

– The instances must have 500GB worth of static dataset that is accessible for the application upon boot up.

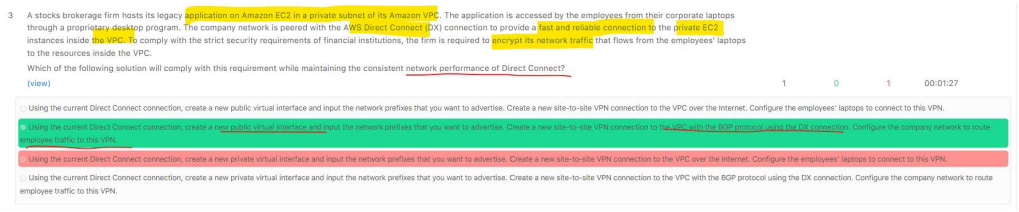
– The instances must be able to scale-out or scale-in depending on the traffic load of the application.

– The Development team must have a quick and automated way to deploy their code updates several times during the day.

– Security patches for the vulnerabilities on the operating system (OS) must be installed within 48 hours of release.

Which of the following solutions should the Solutions Architect implement to meet the company requirements while being cost-effective?

- Install OS patches and create a new AMI using AWS Systems Manager. Use this new AMI for the Auto Scaling group of EC2 instances and replace the existing instances. Deploy the new version of the application to the instances using AWS CodeDeploy. Mount an Amazon EFS volume containing the static dataset on the instances upon boot up.
- Install OS patches and create a new AMI using AWS Systems Manager. Use this new AMI for the Auto Scaling group of EC2 instances and replace the existing instances. Create a scheduled batch job that will run every night to deploy the new application version and install the OS patches. Mount an Amazon EFS volume containing the static dataset on the instances upon boot up.
- Create an Auto Scaling group of EC2 instances using the Amazon Linux AMI. Install the application on the EC2 instances. Replace the existing instances as soon as AWS releases a new Amazon Linux AMI version. Write a user data script that will download the 500 GB static dataset from an Amazon S3 bucket. Deploy the new version of the application to the instances using AWS CodeDeploy.
- Create an Auto Scaling group of EC2 instances using the Amazon Linux AMI. Install the application on the EC2 instances. Write a user data script that will download the 500 GB static dataset from an Amazon S3 bucket. Use AWS Systems Manager to install the OS patches as soon as they are released. Deploy the new version of the application to the instances using AWS CodeDeploy.



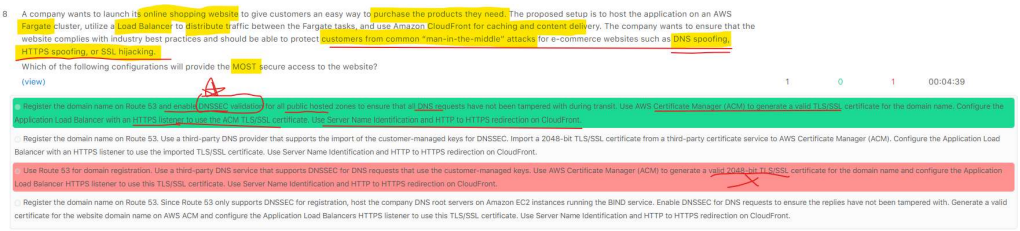
To connect to services such as EC2 using just Direct Connect you need to create a private virtual interface.

However, if you want to encrypt the traffic flowing through Direct Connect, you will need to use the public virtual interface of DX to create a VPN connection that will allow access to AWS services such as S3, EC2, and other services.

To connect to AWS resources that are reachable by a public IP address (such as an Amazon Simple Storage Service bucket) or AWS public endpoints, use a **public virtual interface**. With a public virtual interface, you can:

- Connect to all AWS public IP addresses globally.
- Create public virtual interfaces in any DX location to receive Amazon's global IP routes.
- Access publicly routable Amazon services in any AWS Region (except for the AWS China Region).

If you want to establish a virtual private network (VPN) connection from your company network to an Amazon Virtual Private Cloud (Amazon VPC) over an AWS Direct Connect (DX) connection, you must use a public virtual interface for your DX connection.

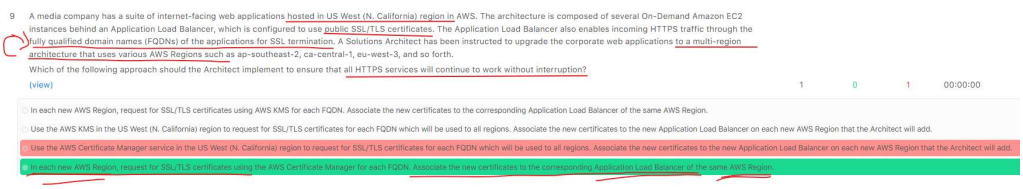


Amazon now allows you to enable **Domain Name System Security Extensions (DNSSEC)** signing for all existing and new public hosted zones, and enable DNSSEC validation for Amazon Route 53 Resolver. **Amazon Route 53 DNSSEC** provides data origin authentication and data integrity verification for DNS and can help customers meet compliance mandates, such as FedRAMP.

> When you enable DNSSEC signing on a hosted zone, Route 53 cryptographically signs each record in that hosted zone. Route 53 manages the zone-signing key, and you can manage the key-signing key in AWS Key Management Service (AWS KMS). Amazon's domain name registrar, Route 53 Domains, already supports DNSSEC, and customers can now register domains and host their DNS on Route 53 with DNSSEC signing enabled. When you enable DNSSEC validation on the Route 53 Resolver in your VPC, it ensures that DNS responses have not been tampered with in transit. This can prevent **DNS Spoofing.**

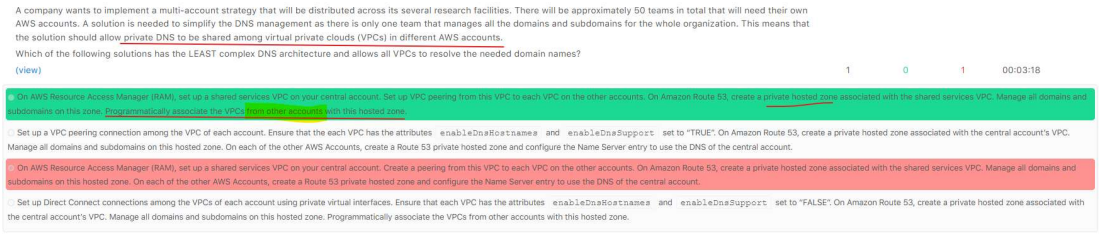
AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the Internet as well as resources on private networks. AWS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates. **Using a valid SSL Certificate for your application load balancer ensures that all requests are encrypted on transit as well as protection against SSL hijacking.**

> CloudFront supports Server Name Indication (SNI) for custom SSL certificates, along with the ability to take incoming H**TTP requests and redirect them to secure HTTPS requests to ensure that clients are always directed to the secure version of your website.**
>
> **Use Route 53 for domain registration. Use a third-party DNS service that supports DNSSEC for DNS requests that use the customer-managed keys. Use AWS Certificate Manager (ACM) to generate a valid 2048-bit TLS/SSL certificate for the domain name and configure the Application Load Balancer HTTPS listener to use this TLS/SSL certificate. Use Server Name Identification and HTTP to HTTPS redirection on CloudFront** is incorrect. This is also possible, but you don't have to rely on a third-party DNS provider as Amazon Route 53 already supports DNSSEC signing.

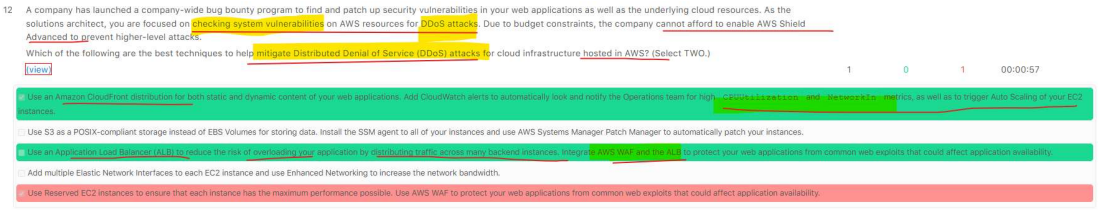


1. To use a certificate with **Elastic Load Balancing** for the same site (the same fully qualified domain name, or FQDN, or set of FQDNs) in a different Region, you **must** request a **new certificate for each Region**
要 每个**region**一个

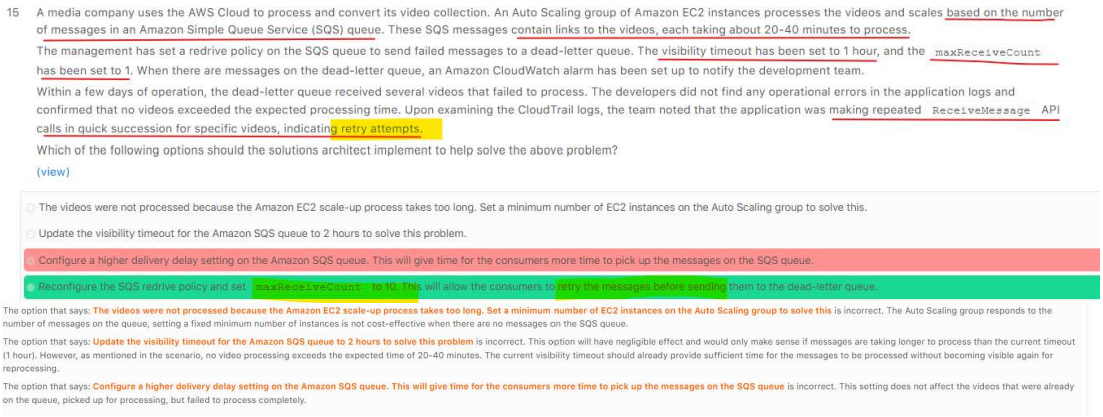
2. To use an ACM certificate with Amazon **CloudFront**, you must request the certificate in the US East (N. Virginia) region. ACM certificates in this region that are associated with a CloudFront distribution are distributed to all the geographic locations configured for that distribution. **host的region 就行 + CloudFront distribution +geographic locations**



You need to associate the VPCs from other accounts to the hosted zone on the central account.



The option that says: **Use Reserved EC2 instances to ensure that each instance has the maximum performance possible. Use AWS WAF to protect your web applications from common web exploits that could affect application availability** is incorrect because using Reserved EC2 instances does not provide any additional computing performance compared to other EC2 types.



重要：Receive Count != Retry Count(SQS 中没有 Retry Count 这个参数！)

Receive count 增加的前提是：Worker successfully received message 但 没有删除（失败 or timeout）
Visibility timeout 到期后消息变可见，可被再次 receive。
所以 receive count = 尝试处理的次数。
ReceiveCount ☑（消息被消费者收到的次数）
maxReceiveCount ☑（超过后进 DLQ）

Delivery Delay（消息投递延迟）— 生产者端的延迟

Delivery delay = delay before the message becomes visible for the first time.

默认：0 秒
最大：15 分钟
效果：
Producer 发送消息 → SQS 故意延迟让消费者看不到
到了 delay 时间之后才会出现在队列里
📌 场景：
你想推迟任务执行，例如 5 秒后/5 分钟后再处理。消费者完全看不到消息直到 delay 结束

Visibility Timeout（可见性超时）— 消费者处理期间的保护时间
Visibility timeout = after a consumer receives a message, SQS hides it temporarily.

默认：30 秒
最大：12 小时

作用：

Consumer 收到消息 → SQS 隐藏此消息
防止其他消费者重复处理

如果 Consumer 在 visibility timeout 内没有删除消息
→ 消息会重新变为可见（被其他消费者处理）

📌 场景：
你的 Lambda/EC2 worker 需要 10 秒来处理消息
设置 visibility=10s → 防止重复处理

AWS Batch Queues enables developers, scientists, and engineers to easily and efficiently run hundreds of thousands of batch computing jobs on AWS. AWS Batch dynamically provisions the optimal quantity and type of compute resources (e.g., CPU or memory optimized instances) based on the volume and specific resource requirements of the batch jobs submitted. With AWS Batch, there is no need to install and manage batch computing software or server clusters that you use to run your jobs, allowing you to focus on analyzing results and solving problems.

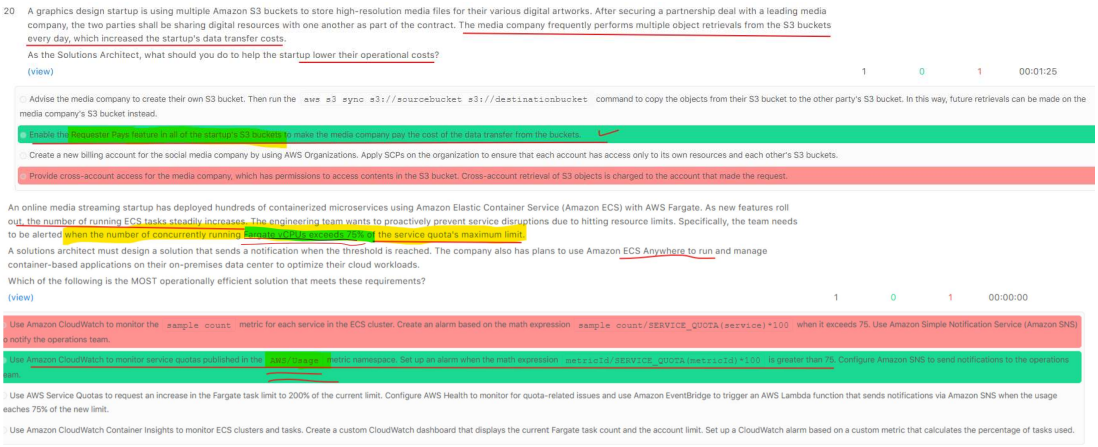
There is **no additional charge for AWS Batch**. You only pay for the AWS resources (e.g. EC2 instances or Fargate jobs) you create to store and run your batch jobs

Compute Environment = AWS Batch 为你创建并管理 EC2 / Fargate 资源的地方

Batch 自动扩缩容（为什么那么强？）
没有任务 → 缩到 0（几乎不花钱）
新任务进入队列 → 自动按需扩容
Spot 不够 → 自动切换 On-Demand
任务完成 → 自动释放多余实例
AWS Batch 是为 不间断、高弹性批处理 专门优化的。

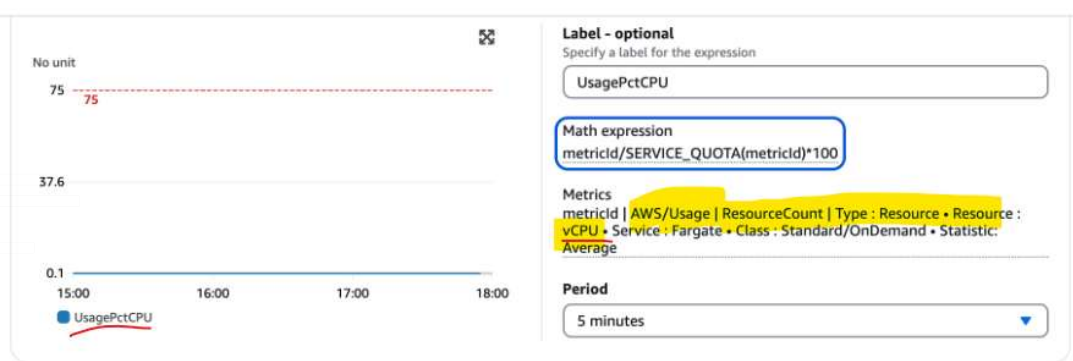
1. 你创建 Job Definition（定义容器、vCPU、Memory）
2. 你创建 Managed Compute Environment（EC2 / Fargate）
3. 你创建 Job Queue，把 CE 绑定进去
4. 提交 Job
5. AWS Batch 自动：
   - 创建 EC2/Fargate
   - 运行容器
   - Job 结束 → 自动缩容

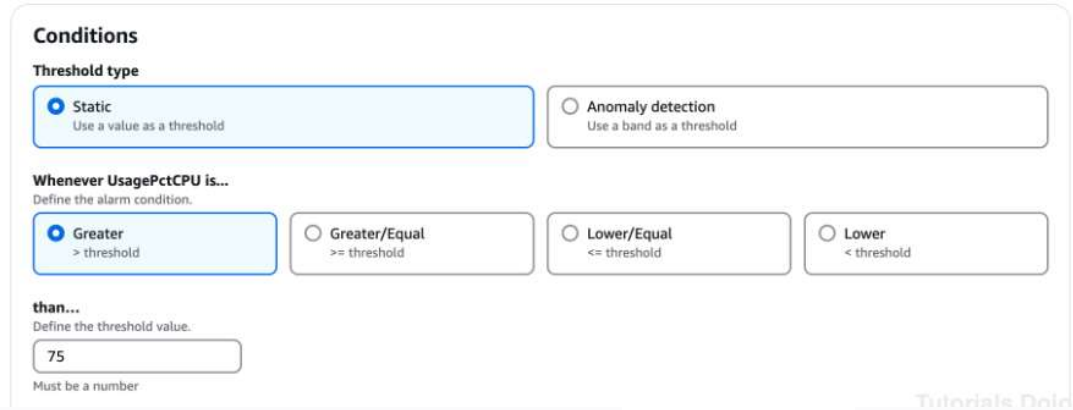
Create an Amazon SQS queue and submit the list of jobs to be processed. Create an Auto Scaling Group of Amazon EC2 Spot Instances that will process the jobs from the SQS queue. Share the raw data across all the instances using Amazon EFS. Store the processed images in an Amazon S3 bucket for long term storage is incorrect as Amazon **EFS is more expensive than storing the raw data on S3 buckets.** This is also **not efficient as listing the jobs on SQS Queue can cause some to be processed twice**, depending on the state of your Spot instances.



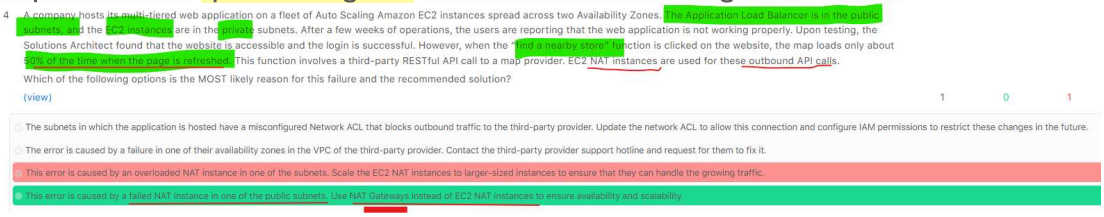
Service Quotas are limits on the **number of resources or operations that can be created or performed within an AWS accoun**t. These quotas exist to prevent accidental over-provisioning and to ensure service stability, and they can often be increased upon request to accommodate growing application needs.

Here, `metricid` is an alias that represents the vCPU count metric in Fargate.



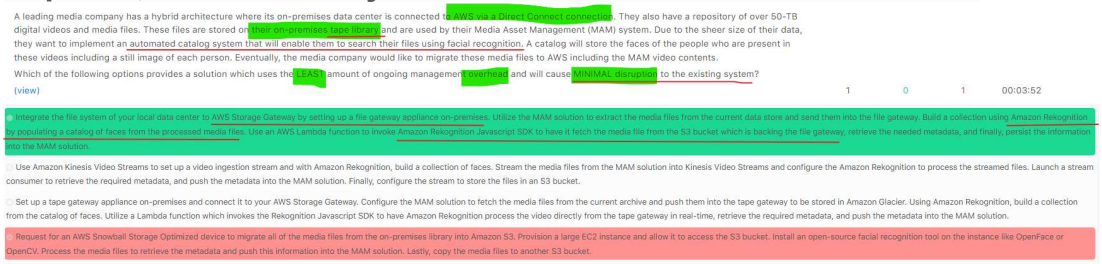


Use Amazon CloudWatch to monitor the `sample count` metric for each service in the ECS cluster. Create an alarm based on the math expression `sample count/SERVICE_QUOTA(service)*100` when it exceeds 75. Use Amazon Simple Notification Service (Amazon SNS) to notify the operations team is incorrect because it primarily focuses on the `sample count` metric, which **monitors the count of requests or invocations in the ECS cluster.** The `sample count` does not directly represent the quota usage or the number of running ECS tasks.



当点击网站上的“查找附近商店”功能时，页面刷新后地图只有大约 50% 的概率能够加载

The option that says: **This error is caused by an overloaded NAT instance in one of the subnets. Scale the EC2 NAT instances to larger-sized instances to ensure that they can handle the growing traffic** is incorrect. If the NAT instances are overloaded, you will typically **notice inconsistent performance or slowdown for the third-party requests.** This failure should have been gone during off-peak hours. If the failure rate is **50% of the requests, it is most likely that one of the NAT instances is down.**



Amazon Rekognition can store information about **detected faces in server-side containers known as collections.** You can use the facial information that's stored in a collection to search for known faces in images, stored videos, and streaming videos. Amazon Rekognition supports the IndexFaces operation. You can use this operation to detect faces in an image and persist information about facial features that are detected in a collection. This is an example of a *storage-based* API operation because the service persists information on the server.
To store facial information, you must first create (CreateCollection) a face collection in one of the AWS Regions in your account. You specify this face collection when you call the `IndexFaces` operation. After you create a face collection and store facial feature information for all faces, you can search the collection for face matches. To search for faces in an image, call SearchFacesByImage. To search for faces in a stored video, call StartFaceSearch. To search for faces in a streaming video, call CreateStreamProcessor.

AWS Storage Gateway offers file-based, volume-based, and tape-based storage solutions. With a tape gateway, you can cost-effectively and durably archive backup data in GLACIER or DEEP_ARCHIVE. A tape gateway provides a virtual tape infrastructure that scales seamlessly with your business needs and eliminates the operational burden of provisioning, scaling, and maintaining a physical tape infrastructure.

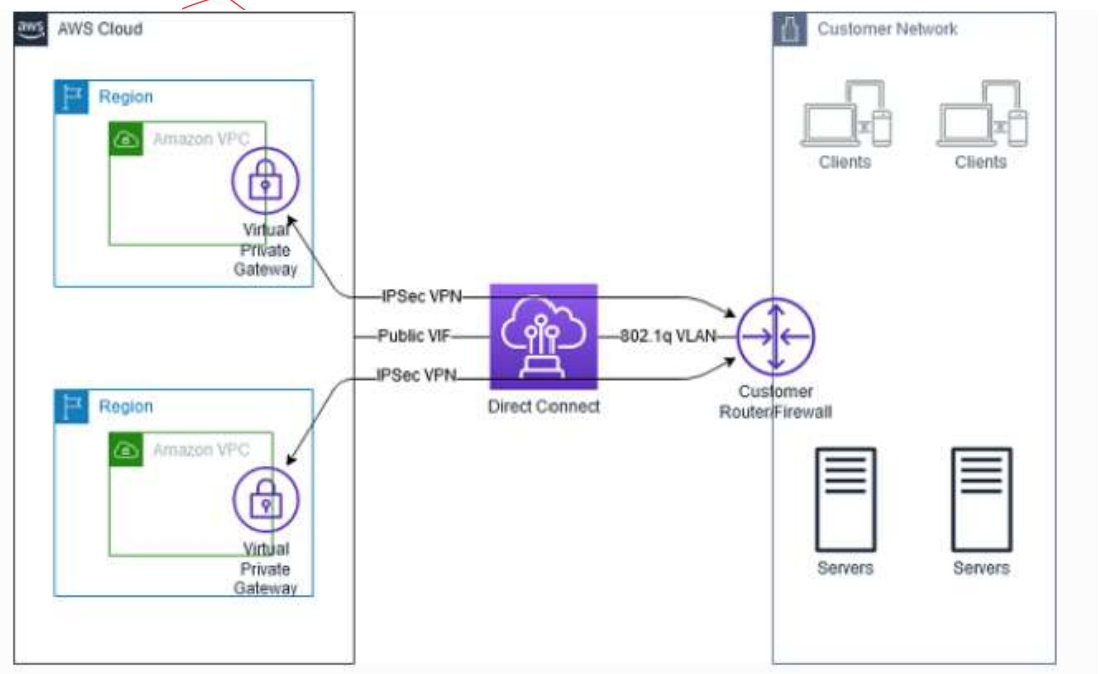
You can run AWS Storage Gateway either on-premises as a VM appliance, as a hardware appliance, or in AWS as an Amazon Elastic Compute Cloud (Amazon EC2) instance. You deploy your gateway on an EC2 instance to provision **iSCSI** storage volumes in AWS. You can use gateways hosted on EC2 instances for disaster recovery, data mirroring, and providing storage for applications hosted on Amazon EC2.

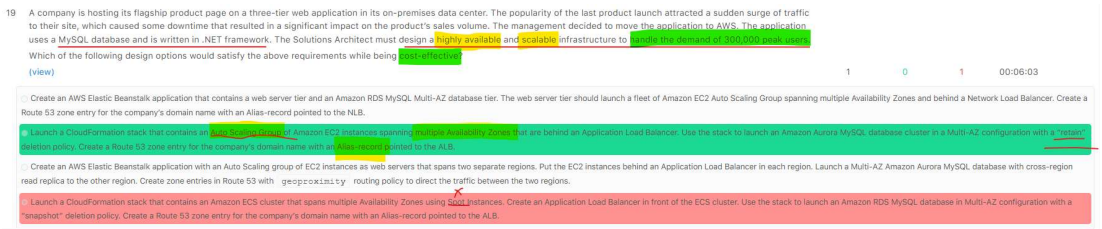
14 A multinational investment bank has a hybrid cloud architecture that uses a single 1 Gbps AWS Direct Connect connection to integrate their on-premises network to AWS Cloud. The bank has a total of 10 VPCs which are all connected to their on-premises data center via the same Direct Connect connection that you manage. Based on the recent IT audit, the existing network setup has a single point of failure which needs to be addressed immediately.

Which of the following is the MOST cost-effective solution that you should implement in order to improve the connection redundancy of your hybrid network?

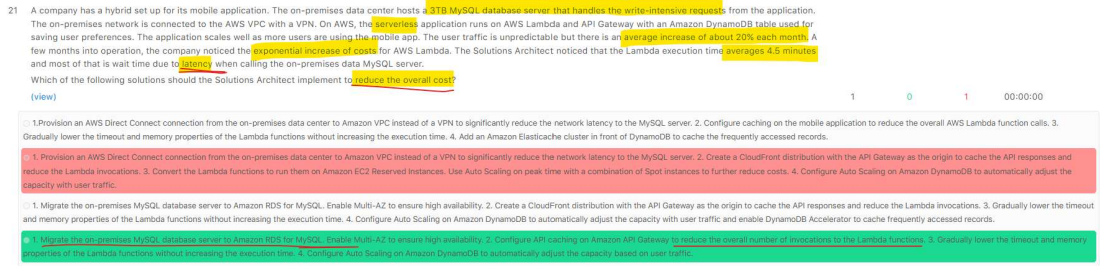
(view) 1 0 1 00:00:00

- Establish another 1 Gbps AWS Direct Connect connection using a public Virtual Interface (VIF). Prepare a VPN tunnel that will terminate on the virtual private gateway (VGW) of the respective VPC using the public VIF. Handle the failover to the VPN connection through the use of BGP.
- Establish a new point-to-point Multiprotocol Label Switching (MPLS) connection to all of your 10 VPCs. Configure BGP to use this new connection with an active/passive routing.
- Establish VPN tunnels from your on-premises data center to each of the 10 VPCs. Terminate each VPN tunnel connection at the virtual private gateway (VGW) of the respective VPC. Configure BGP for route management.
- Establish another 1 Gbps AWS Direct Connect connection with corresponding private Virtual Interfaces (VIFs) to connect all of the 10 VPCs individually. Set up a Border Gateway Protocol (BGP) peering session for all of the VIFs.





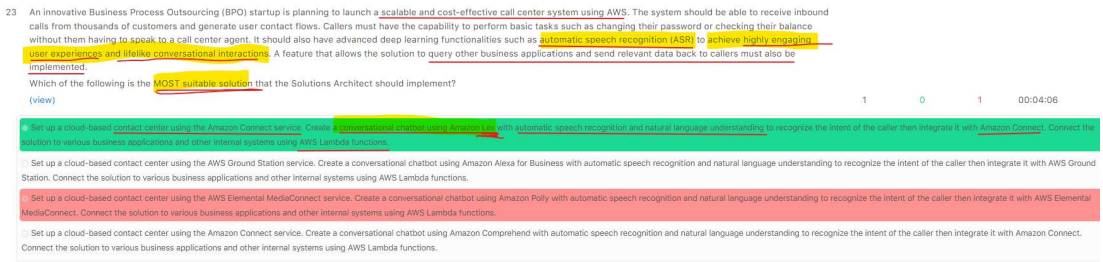
The option that says: **Launch a CloudFormation stack that contains an Amazon ECS cluster that spans multiple Availability Zones using Spot Instances. Create an Application Load Balancer in front of the ECS cluster. Use the stack to launch an Amazon RDS MySQL database in Multi-AZ configuration with a "snapshot" deletion policy. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the ALB** is incorrect. Although the Spot instance provides good cost savings for the web tier, the reliability of the site will suffer as the Spot instances are usually r**eclaimed by AWS based on the supply and demand of its global computing capacit**y. The "snapshot" deletion policy on the database tier is also not ideal as this will require a significant time to restore if you delete the CloudFormation stack.



The following option is incorrect:

- Provision an AWS Direct Connect connection from the on-premises data center to Amazon VPC instead of a VPN to significantly reduce the network latency to the MySQL server.
- Create a CloudFront distribution with the API Gateway as the origin to cache the API responses and reduce the Lambda invocations.
- Convert the Lambda functions to run them on Amazon EC2 Reserved Instances. Use Auto Scaling on peak time with a combination of Spot instances to further reduce costs.
- Configure Auto Scaling on Amazon DynamoDB to automatically adjust the capacity with user traffic.

Provisioning a Direct Connection just for the application is not economical even if it offers better latency than a VPN connection. Caching the API requests should be done on the API Gateway, and not on CloudFront. EC2 Reserve instances could be more expensive than Lambda functions when application traffic is low.

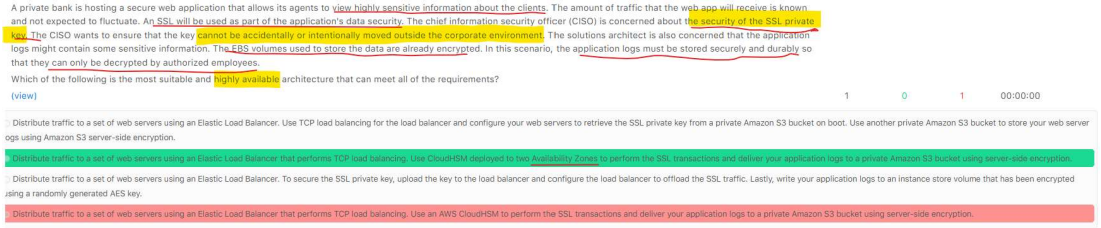


Amazon Connect provides a seamless omnichannel 全渠道experience through a single unified contact center for voice and chat. Contact center

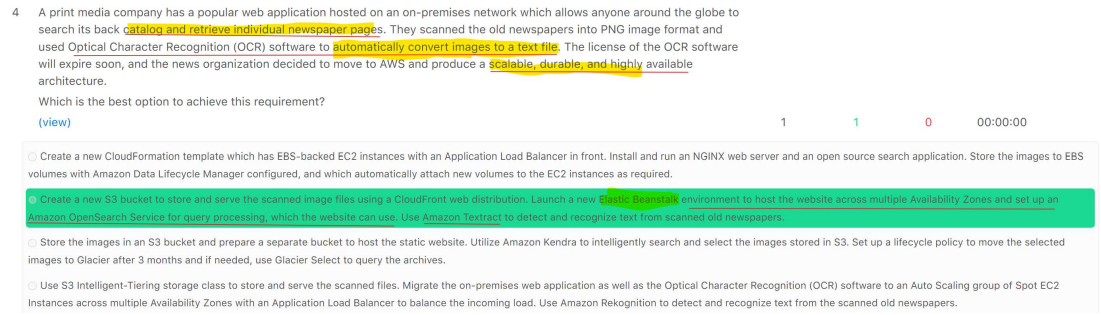
agents and managers don't have to learn multiple tools because **Amazon Connect has the same contact routing, queuing, analytics, and management tools in a single UI across voice, web chat, and mobile chat.**

Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of **automatic speech recognition (ASR)** for converting speech to text and natural language understanding (NLU) to recognize the intent of the text, enabling you to build applications with **highly engaging user experiences and lifelike conversational interactions.** With Amazon Lex, the same deep learning technologies that power Amazon Alexa are now available to any developer, enabling you to quickly and easily build sophisticated, natural language conversational bots ("chatbots").

The option that says: **Set up a cloud-based contact center using the AWS Elemental MediaConnect service. Create a conversational chatbot using Amazon Polly with automatic speech recognition and natural language understanding to recognize the intent of the caller then integrate it with AWS Elemental MediaConnect. Connect the solution to various business applications and other internal systems using AWS Lambda functions** is incorrect because AWS Elemental **MediaConnect is just a high-quality transport service for live video.** You have to use Amazon Connect instead to set up your cloud-based contact center. And although **Amazon Polly is a machine learning service,** it is quite limited as it just turns text into lifelike speech that allows you to create applications that talk and build entirely new categories of speech-enabled products. A more suitable service to use here is Amazon Lex.

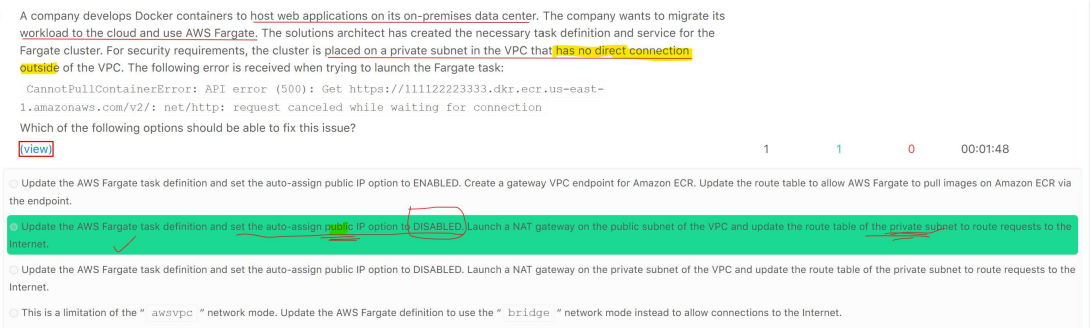


The option that says: **Distribute traffic to a set of web servers using an Elastic Load Balancer that performs TCP load balancing. Use an AWS CloudHSM to perform the SSL transactions and deliver your application logs to a private Amazon S3 bucket using server-side encryption** is incorrect. Although it is almost similar to the correct option, the a**rchitecture did not explicitly say that the CloudHSM is deployed to multiple Availability Zones,** which means that this architecture is **not highly available** compared with the correct option.



Amazon OpenSearch Service simplifies the deployment, operation, and scaling of OpenSearch, a widely used open-source search and analytics

engine. It provides robust security features, ensures high availability and data durability, and offers direct access to the OpenSearch API.
With Amazon OpenSearch Service, you can quickly add rich search capabilities to your website or application. You don't need to become a search expert or worry about hardware provisioning, setup, and maintenance. With a few clicks in the AWS Management Console, you can create a search domain and upload the data that you want to make searchable, and Amazon OpenSearch Service will automatically provision the required resources and deploy a highly-tuned search index.
You can easily change your search parameters, fine-tune search relevance, and apply new settings at any time. As your volume of data and traffic fluctuates, Amazon OpenSearch Service seamlessly scales to meet your needs.


9 A company develops Docker containers to host web applications on its on-premises data center. The company wants to migrate its workload to the cloud and use AWS Fargate. The solutions architect has created the necessary task definition and service for the Fargate cluster. For security requirements, the cluster is placed on a private subnet in the VPC that has no direct connection outside of the VPC. The following error is received when trying to launch the Fargate task:

CannotPullContainerError: API error (500): Get https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: request canceled while waiting for connection

Which of the following options should be able to fix this issue?

(view) 1 1 0 00:01:48

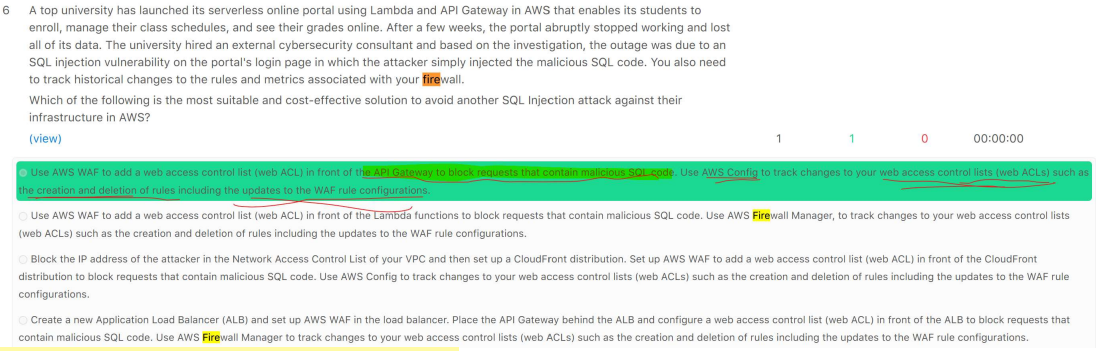
Update the AWS Fargate task definition and set the auto-assign public IP option to ENABLED. Create a gateway VPC endpoint for Amazon ECR. Update the route table to allow AWS Fargate to pull images on Amazon ECR via the endpoint.

Update the AWS Fargate task definition and set the auto-assign public IP option to DISABLED. Launch a NAT gateway on the public subnet of the VPC and update the route table of the private subnet to route requests to the Internet.

Update the AWS Fargate task definition and set the auto-assign public IP option to DISABLED. Launch a NAT gateway on the private subnet of the VPC and update the route table of the private subnet to route requests to the Internet.

This is a limitation of the " awsvpc " network mode. Update the AWS Fargate definition to use the " bridge " network mode instead to allow connections to the Internet.

The **`CannotPullContainer error (500)`** is caused by the `Connection timed out` when connecting to Amazon ECR. This indicates that when creating a task, the container image specified could not be retrieved.
When a Fargate task is launched, its elastic network interface requires a route to the Internet to pull container images. If you receive an error similar to the following when launching a task, it is because a route to the Internet does not exist:
`CannotPullContainerError: API error (500): Get https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: request canceled while waiting for connection"`
To resolve this issue, you can:
- For tasks in public subnets, specify **ENABLED** for **Auto-assign public IP** when launching the task.
- For tasks in private subnets, specify **DISABLED** for **Auto-assign public IP** when launching the task, and configure a NAT gateway in your VPC to route requests to the Internet.


6 A top university has launched its serverless online portal using Lambda and API Gateway in AWS that enables its students to enroll, manage their class schedules, and see their grades online. After a few weeks, the portal abruptly stopped working and lost all of its data. The university hired an external cybersecurity consultant and based on the investigation, the outage was due to an SQL injection vulnerability on the portal's login page in which the attacker simply injected the malicious SQL code. You also need to track historical changes to the rules and metrics associated with your firewall.

Which of the following is the most suitable and cost-effective solution to avoid another SQL Injection attack against their infrastructure in AWS?

(view) 1 1 0 00:00:00

Use AWS WAF to add a web access control list (web ACL) in front of the API Gateway to block requests that contain malicious SQL code. Use AWS Config to track changes to your web access control lists (web ACLs) such as the creation and deletion of rules including the updates to the WAF rule configurations.

Use AWS WAF to add a web access control list (web ACL) in front of the Lambda functions to block requests that contain malicious SQL code. Use AWS Firewall Manager, to track changes to your web access control lists (web ACLs) such as the creation and deletion of rules including the updates to the WAF rule configurations.

Block the IP address of the attacker in the Network Access Control List of your VPC and then set up a CloudFront distribution. Set up AWS WAF to add a web access control list (web ACL) in front of the CloudFront distribution to block requests that contain malicious SQL code. Use AWS Config to track changes to your web access control lists (web ACLs) such as the creation and deletion of rules including the updates to the WAF rule configurations.

Create a new Application Load Balancer (ALB) and set up AWS WAF in the load balancer. Place the API Gateway behind the ALB and configure a web access control list (web ACL) in front of the ALB to block requests that contain malicious SQL code. Use AWS Firewall Manager to track changes to your web access control lists (web ACLs) such as the creation and deletion of rules including the updates to the WAF rule configurations.

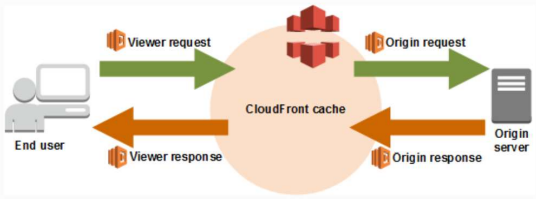
malicious SQL code恶意SQL代码

14. QUESTION

An international foreign exchange company has a serverless forex trading application that was built using AWS SAM and is hosted on AWS Serverless Application Repository. They have millions of users worldwide who use their online portal 24/7 to trade currencies. However, they are receiving a lot of complaints that it takes a few minutes for their users to log in to their portal lately, including occasional HTTP 504 errors. As the Solutions Architect, you are tasked to optimize the system and to significantly reduce the time to log in to improve the customers' satisfaction.

Which of the following should you implement in order to improve the performance of the application with minimal cost? (Select TWO.)

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



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

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

25. QUESTION

A multinational financial company has a suite of web applications hosted in multiple VPCs in various AWS regions. As part of their security compliance, the company's Solutions Architect has been tasked to set up a logging solution to track all of the changes made to their AWS resources in all regions, which host their enterprise accounting systems. The company is using different AWS services such as Amazon EC2 instances, Amazon S3 buckets, CloudFront web distributions, and AWS IAM. The logging solution must ensure the security, integrity, and durability of your log data in order to pass the compliance requirements. In addition, it should provide an event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command-line tools, and API calls.

In this scenario, which of the following options is the best solution to use?

- ( ) Create a new AWS CloudTrail trail in a new S3 bucket using the AWS CLI and also pass the –no-include-global-service-events and –is-multi-region-trail parameter then encrypt log files using KMS encryption. Enable Multi-Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- ( ) Create a new Amazon CloudWatch trail in a new S3 bucket using the AWS CLI and also pass both the –is-multi-region-trail and –include-global-service-events parameters then encrypt log files using KMS encryption. Enable Multi-Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- (•) Create a new Amazon CloudWatch trail in a new S3 bucket using the AWS CLI and also pass the –include-global-service-events parameter then encrypt log files using KMS encryption. Enable Multi-Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- ( ) Create a new AWS CloudTrail trail in a new S3 bucket using the AWS CLI and also pass both the –is-multi-region-trail and –include-global-service-events parameters then encrypt log files using KMS encryption. Enable Multi-Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.

38. QUESTION

A leading financial company is planning to launch its Node.js application with an Amazon RDS MariaDB database to serve its clients worldwide. The application will run on both on-premises servers as well as Reserved EC2 instances. To comply with the company's strict security policy, the database credentials must be encrypted both at rest and in transit. These credentials will be used by the application servers to connect to the database. The Solutions Architect is tasked to manage all of the aspects of the application architecture and production deployment.

How should the Architect automate the deployment process of the application in the MOST secure manner?

- Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Attach this IAM policy to the instance profile for CodeDeploy-managed EC2 instances. Associate the same policy as well to the on-premises instances. Using AWS CodeDeploy, launch the application packages to the Amazon EC2 instances and on-premises servers.
- Upload the database credentials with key rotation in AWS Secrets Manager. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to all on-premises servers and EC2 instances. Use Elastic Beanstalk to host and manage the application on both on-premises servers and EC2 instances. Deploy the succeeding application revisions to AWS and on-premises servers using Elastic Beanstalk.
- Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to the EC2 instances. Create an IAM Service Role that will be associated with the on-premises servers. Deploy the application packages to the EC2 instances and on-premises servers using AWS CodeDeploy.
- Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to all on-premises servers and EC2 instances. Use Elastic Beanstalk to host and manage the application on both on-premises servers and EC2 instances. Deploy the succeeding application revisions to AWS and on-premises servers using Elastic Beanstalk.

The option that says: **Upload the database credentials with a Secure String data type in AWS Systems Manager Parameter Store. Install the AWS SSM agent on all servers. Set up a new IAM role that enables access and decryption of the database credentials from SSM Parameter Store. Associate this role to all on-premises servers and EC2 instances. Use Elastic Beanstalk to host and manage the application on both on-premises servers and EC2 instances. Deploy the succeeding application revisions to AWS and on-premises servers using Elastic Beanstalk** is incorrect. You can't deploy an application to your on-premises servers using Elastic Beanstalk. This is only applicable to your Amazon EC2 instances.

Service role: A service role is an AWS Identity and Access Management (IAM) that grants permissions to an AWS service so that the service can access AWS resources. Only a few Systems Manager scenarios require a service role. When you create a service role for Systems Manager, you choose the permissions to grant in order for it to access or interact with other AWS resources.
Service-linked role: A service-linked role is predefined by Systems Manager and includes all the permissions that the service requires to call other AWS services on your behalf.

42. QUESTION

A company is using AWS Organizations to manage their multi-account and multi-region AWS infrastructure. They are currently doing large-scale automation for their key daily processes to save costs. One of these key processes is sharing specified AWS resources, which an organizational account owns, with other AWS accounts of the company using AWS RAM. There is already an existing service which was previously managed by a separate organization account moderator, who also maintained the specific configuration details.

In this scenario, what could be a simple and effective solution that would allow the service to perform its tasks on the organization accounts on the moderator's behalf?

- Attach an IAM role on the service detailing all the allowed actions that it will be able to perform. Install an SSM agent in each of the worker VMs. Use AWS Systems Manager to build automation workflows that involve the daily key processes.
- Use trusted access by running the `enable-sharing-with-aws-organization` command in the AWS RAM CLI. Mirror the configuration changes that was performed by the account that previously managed this service.
- Enable cross-account access with AWS Organizations in the Resource Access Manager Console. Mirror the configuration changes that was performed by the account that previously managed this service.
- Configure a service-linked role for AWS RAM and modify the permissions policy to specify what the role can and cannot do. Lastly, modify the trust policy of the role so that other processes can utilize AWS RAM.

AWS Resource Access Manager (AWS RAM) enables you to share specified AWS resources that you own with other AWS accounts. To enable trusted access with AWS Organizations:

1. From the AWS RAM CLI, use the `enable-sharing-with-aws-organizations` command.
2. Name of the IAM service-linked role that can be created in accounts when trusted access is enabled: *AWSResourceAccessManagerServiceRolePolicy*.
3. **Primeiro você habilita o Trusted Access** *“enable-sharing-with-aws-organization”* **(CLI), depois** *Enable cross-account access with AWS Organizations” (no Console do RAM)*

☑ 1 Primeiro:

Habilitar Trusted Access

📌 *“enable-sharing-with-aws-organization” (via CLI)*

Isso **ativa a integração** entre o AWS RAM e o AWS Organizations.

Sem isso, **nenhum recurso pode ser compartilhado automaticamente** entre contas da Organization.

👉 **É uma configuração global da Organization.**

👉 Feita **uma vez só**, normalmente pelo *management account*.

☑ 2 Depois:

Compartilhar o recurso

📌 *“Enable cross-account access with AWS Organizations” (no Console do RAM)*

Agora que o Trusted Access está ligado, você pode:

- compartilhar uma VPC Subnet
- compartilhar uma private hosted zone
- compartilhar um cluster Aurora
- compartilhar um KMS key grant
- etc.

Escolhendo:

- **toda a Organization**,
- **uma OU específica**,

- **contas específicas**.

43. QUESTION

A company processes several petabytes of images submitted by users on their photo hosting site every month. Each month, the images are processed in its on-premises data center by a High-Performance Computing (HPC) cluster with a capacity of 5,000 cores and 10 petabytes of data. Processing a month's worth of images by thousands of jobs running in parallel takes about a week and the processed images are stored on a network file server, which also backups the data to a disaster recovery site.

The current data center is nearing its capacity so the users are forced to spread the jobs within the course of the month. This is not ideal for the requirement of the jobs, so the Solutions Architect was tasked to design a scalable solution that can exceed the current capacity with the least amount of management overhead while maintaining the current level of durability.

Which of the following solutions will meet the company's requirements while being cost-effective?

- Package the executable file for the job in a Docker image stored on Amazon Elastic Container Registry (Amazon ECR). Run the Docker images on Amazon Elastic Kubernetes Service (Amazon EKS). Auto Scaling can be handled automatically by EKS. Store the raw data temporarily on Amazon EBS SC1 volumes and then send the images to an Amazon S3 bucket after processing.
- Utilize AWS Batch with Managed Compute Environments to create a fleet using Spot Instances. Store the raw data on an Amazon S3 bucket. Create jobs on AWS Batch Job Queues that will pull objects from the Amazon S3 bucket and temporarily store them to the EC2 EBS volumes for processing. Send the processed images back to another Amazon S3 bucket.
- Using a combination of On-demand and Reserved Instances as Task Nodes, create an EMR cluster that will use Spark to pull the raw data from an Amazon S3 bucket. List the jobs that need to be processed by the EMR cluster on a DynamoDB table. Store the processed images on a separate Amazon S3 bucket.
- Create an Amazon SQS queue and submit the list of jobs to be processed. Create an Auto Scaling Group of Amazon EC2 Spot Instances that will process the jobs from the SQS queue. Share the raw data across all the instances using Amazon EFS. Store the processed images in an Amazon S3 bucket for long term storage.

The option that says: **Create an Amazon SQS queue and submit the list of jobs to be processed. Create an Auto Scaling Group of Amazon EC2 Spot Instances that will process the jobs from the SQS queue. Share the raw data across all the instances using Amazon EFS. Store the processed images in an Amazon S3 bucket for long term storage** is incorrect as Amazon EFS is more expensive than storing the raw data on S3 buckets. This is also not efficient as listing the jobs on SQS Queue can **cause some to be processed twice, depending on the state of your Spot instances**

64. QUESTION

A stocks brokerage firm hosts its legacy application on Amazon EC2 in a private subnet of its Amazon VPC. The application is accessed by the employees from their corporate laptops through a proprietary desktop program. The company network is peered with the AWS Direct Connect (DX) connection to provide a fast and reliable connection to the private EC2 instances inside the VPC. To comply with the strict security requirements of financial institutions, the firm is required to encrypt its network traffic that flows from the employees' laptops to the resources inside the VPC.

Which of the following solution will comply with this requirement while maintaining the consistent network performance of Direct Connect?

- Using the current Direct Connect connection, create a new public virtual interface and input the network prefixes that you want to advertise. Create a new site-to-site VPN connection to the VPC over the Internet. Configure the employees' laptops to connect to this VPN.
- Using the current Direct Connect connection, create a new private virtual interface and input the network prefixes that you want to advertise. Create a new site-to-site VPN connection to the VPC over the Internet. Configure the employees' laptops to connect to this VPN.
- Using the current Direct Connect connection, create a new private virtual interface and input the network prefixes that you want to advertise. Create a new site-to-site VPN connection to the VPC with the BGP protocol using the DX connection. Configure the company network to route employee traffic to this VPN.
- Using the current Direct Connect connection, create a new public virtual interface and input the network prefixes that you want to advertise. Create a new site-to-site VPN connection to the VPC with the BGP protocol using the DX connection. Configure the company network to route employee traffic to this VPN.

If you want to establish a **virtual private network (VPN)** connection from your **company network** to an Amazon Virtual Private Cloud (Amazon **VPC**) over an

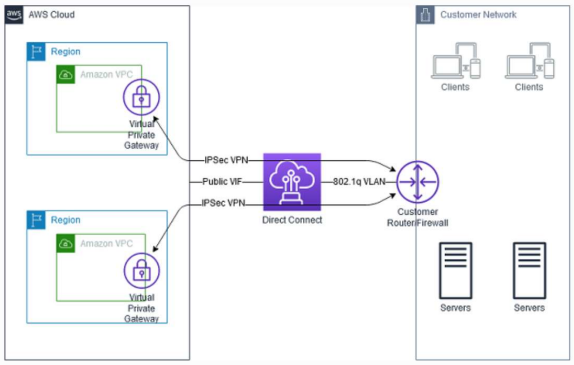
AWS Direct Connect (DX) connection, you must use a **public virtual interface** for your DX connection.

To connect to AWS resources that are reachable by a public IP address (such as an Amazon Simple Storage Service bucket) or AWS public endpoints, use a **public virtual interface**. With a public virtual interface, you can:

– Connect to all AWS public IP addresses globally.

– Create public virtual interfaces in any DX location to receive Amazon's global IP routes.

– Access publicly routable Amazon services in any AWS Region (except for the AWS China Region).



39. QUESTION

A company needs a deployment solution for its application that is hosted on the AWS cloud. The company has the following requirements for the application:

– The instances must have 500GB worth of static dataset that is accessible for the application upon boot up.

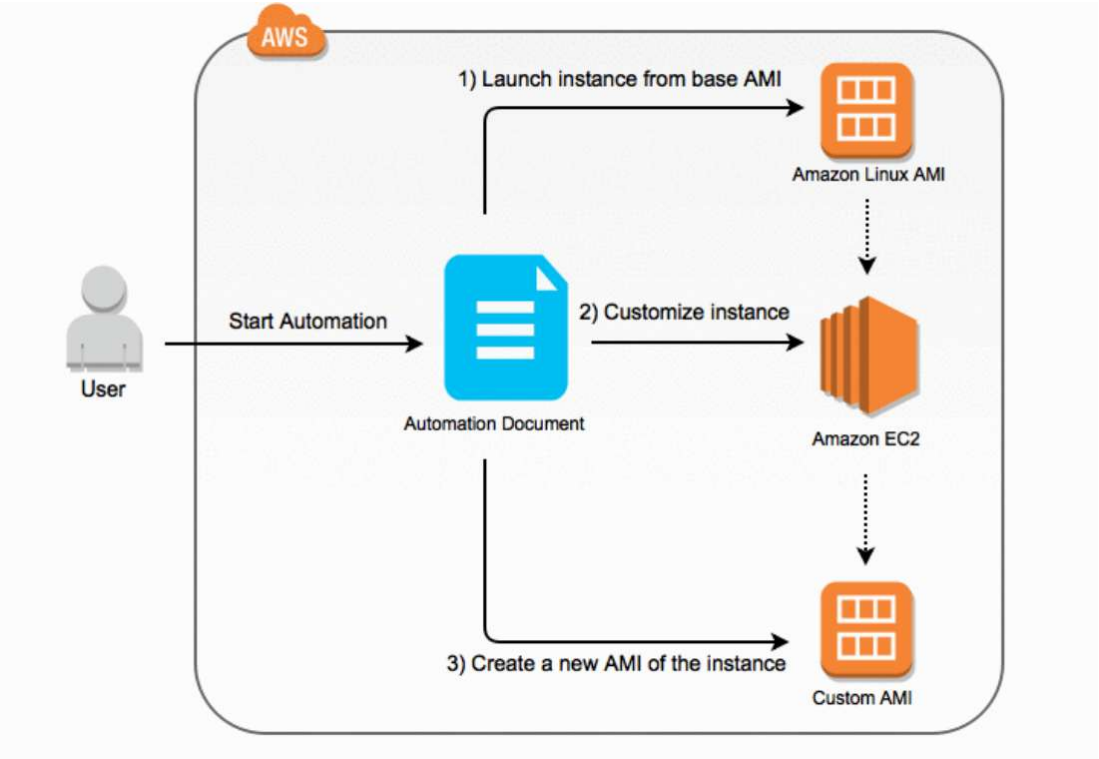
– The instances must be able to scale-out or scale-in depending on the traffic load of the application.

– The Development team must have a quick and automated way to deploy their code updates several times during the day.

– Security patches for the vulnerabilities on the operating system (OS) must be installed within 48 hours of release.

Which of the following solutions should the Solutions Architect implement to meet the company requirements while being cost-effective?

- Create an Auto Scaling group of EC2 instances using the Amazon Linux AMI. Install the application on the EC2 instances. Write a user data script that will download the 500 GB static dataset from an Amazon S3 bucket. Use AWS Systems Manager to install the OS patches as soon as they are released. Deploy the new version of the application to the instances using AWS CodeDeploy.
- Install OS patches and create a new AMI using AWS Systems Manager. Use this new AMI for the Auto Scaling group of EC2 instances and replace the existing instances. Create a scheduled batch job that will run every night to deploy the new application version and install the OS patches. Mount an Amazon EFS volume containing the static dataset on the instances upon boot up.
- Create an Auto Scaling group of EC2 instances using the Amazon Linux AMI. Install the application on the EC2 instances. Replace the existing instances as soon as AWS releases a new Amazon Linux AMI version. Write a user data script that will download the 500 GB static dataset from an Amazon S3 bucket. Deploy the new version of the application to the instances using AWS CodeDeploy.
- **Install OS patches and create a new AMI using AWS Systems Manager. Use this new AMI for the Auto Scaling group of EC2 instances and replace the existing instances. Deploy the new version of the application to the instances using AWS CodeDeploy. Mount an Amazon EFS volume containing the static dataset on the instances upon boot up.**



The option that says: **Create an Auto Scaling group of EC2 instances using the Amazon Linux AMI. Install the application on the EC2 instances. Write a user data script that will download the 500 GB static dataset from an Amazon S3 bucket. Use AWS Systems Manager to install the OS patches as soon as they are released. Deploy the new version of the application to the instances using AWS CodeDeploy** is incorrect. Although Amazon S3 may seem more cost-effective than Amazon EFS in storing static contents, the Amazon EC2 instances **will have to download the dataset on its local EBS volume. Attaching 500GB EBS volumes on each of the EC2 instances is more expensive compared to just using a single EFS volume mounted on all EC2 instances** at boot up.

41. QUESTION

A company is modernizing its on-premises system by migrating it to AWS. The system will be hosted on EC2 instances managed by Amazon Elastic Kubernetes Service (EKS) and use Amazon RDS for MySQL as the database. The system has predictable schedules of high usage, especially during sales events and holiday seasons.

What pricing options should the company consider when selecting the MOST cost-optimized solution?

- Acquire EC2 Instance Savings Plans for the EC2 nodes of the EKS cluster to be used for regular traffic. Scale the node cluster with On-Demand Instances during peak demands. Handle the predicted database load with a 1-year Partial Upfront Reserved Instance and vertically scale up the DB instance on scheduled high usage.
- Purchase Compute Savings Plans for the EC2 nodes of the EKS cluster to be used for regular traffic. Scale the node cluster with Spot instances during peak demands. Handle the predicted database load with a 1-year All Upfront Reserved Instance and vertically scale up the DB instance on scheduled high usage.
- Acquire Compute Savings Plans for the EC2 nodes of the EKS cluster to be used for regular traffic. Scale the node cluster with On-Demand Capacity Reservations during peak demands. Handle the predicted database load with a 1-year No Upfront Reserved Instance and increase database read replicas on scheduled high usage.
- Purchase Standard Reserved Instances for the EC2 nodes of the EKS cluster to be used for regular traffic. Scale the node cluster with Dedicated Instances during peak demands. Handle the predicted database load with a 1-year All Upfront Reserved Instance.

For EC2 usage, EC2 Instance Savings Plans offer the highest savings (up to 72%) and are **applied to a specific instance within a chosen region.** Compute Savings Plans offer slightly lower savings (up to 66%) but **provide more flexibility as they apply to any instance family and can cover usage across different services like EC2, Fargate, and Lambda.** For use cases when there is an increase in load during certain periods, Spot instances can be a cost-effective

solution. These instances provide flexibility and allow using spare EC2 capacity at a significant discount.

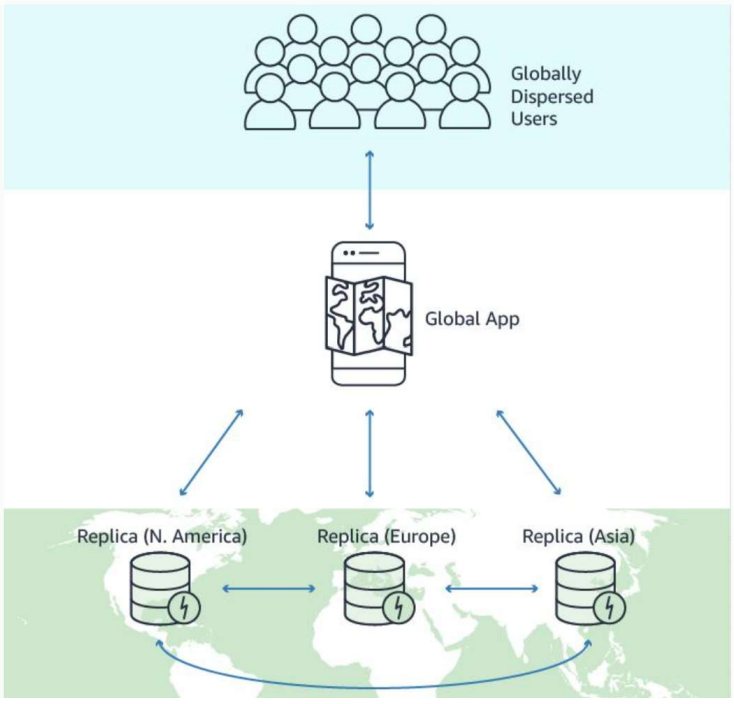
The option that says: **Acquire EC2 Instance Savings Plans for the EC2 nodes of the EKS cluster to be used for regular traffic. Scale the node cluster with On-Demand Instances during peak demands. Handle the predicted database load with a 1-year Partial Upfront Reserved Instance and vertically scale up the DB instance on scheduled high usage** is incorrect. While EC2 Instance Savings Plans do offer a discount over On-Demand pricing, these only apply to a specific instance family and will limit the flexibility of changing instance types. This option suggests scaling with On-Demand Instances, which are more expensive than Spot instances. Moreover, using a 1-year Partial Upfront Reserved Instance for the database can offer some cost savings, but not as much savings as an All Upfront Reserved Instance.

44. QUESTION

A tech startup is planning to launch a new global mobile marketplace using AWS Amplify and AWS Mobile Hub. To lower the latency, the backend APIs will be launched to multiple AWS regions to process the sales and financial transactions in the region closest to the users. The solutions architect is instructed to design the system architecture to ensure that the transactions made in one region are automatically replicated to other regions. In the coming months ahead, it is expected that the marketplace will have millions of users across North America, South America, Europe, and Asia.

Which of the following is the most scalable, cost-effective, and highly available architecture that you should implement?

- Use a combination of AWS Control Tower and Amazon Connect to launch and centrally manage multiple DynamoDB tables in various AWS Regions. In each local region, store the individual transactions to a DynamoDB replica table in the same region.
- Create Amazon S3 buckets in all required regions. Store the individual transactions in the S3 bucket in the local region. Replicate the transactions between regions using S3 Cross-Region Replication.
- In each local region, store the individual transactions to a DynamoDB table. Set up an AWS Lambda function to read recent writes from the table, and replay the data to DynamoDB tables in all other regions.
- Create a Global DynamoDB table with replica tables across several AWS regions that you prefer. In each local region, store the individual transactions to a DynamoDB replica table in the same region. Any changes made in one of the replica tables will automatically be replicated across all other tables.



45. QUESTION

A company is hosting its flagship product page on a three-tier web application in its on-premises data center. The popularity of the last product launch attracted a sudden surge of traffic to their site, which caused some downtime that resulted in a significant impact on the product's sales volume. The management decided to move the application to AWS. The application uses a MySQL database and is written in .NET framework. The Solutions Architect must design a highly available and scalable infrastructure to handle the demand of 300,000 peak users.

Which of the following design options would satisfy the above requirements while being cost-effective?

- Create an AWS Elastic Beanstalk application with an Auto Scaling group of EC2 instances as web servers that spans two separate regions. Put the EC2 instances behind an Application Load Balancer in each region. Launch a Multi-AZ Amazon Aurora MySQL database with cross-region read replica to the other region. Create zone entries in Route 53 with `geoproximity` routing policy to direct the traffic between the two regions.
- Launch a CloudFormation stack that contains an Amazon ECS cluster that spans multiple Availability Zones using Spot Instances. Create an Application Load Balancer in front of the ECS cluster. Use the stack to launch an Amazon RDS MySQL database in Multi-AZ configuration with a "snapshot" deletion policy. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the ALB.
- Launch a CloudFormation stack that contains an Auto Scaling Group of Amazon EC2 instances spanning multiple Availability Zones that are behind an Application Load Balancer. Use the stack to launch an Amazon Aurora MySQL database cluster in a Multi-AZ configuration with a "retain" deletion policy. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the ALB.
- Create an AWS Elastic Beanstalk application that contains a web server tier and an Amazon RDS MySQL Multi-AZ database tier. The web server tier should launch a fleet of Amazon EC2 Auto Scaling Group spanning multiple Availability Zones and behind a Network Load Balancer. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the NLB.

The option that says: **Create an AWS Elastic Beanstalk application that contains a web server tier and an Amazon RDS MySQL Multi-AZ database tier. The web server tier should launch a fleet of Amazon EC2 Auto Scaling Group spanning multiple Availability Zones and behind a Network Load Balancer. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the NLB** is incorrect. Network Load Balancer (NLB) is typically used for TCP traffic (Layer 4), and is not suitable for handling web applications that work on HTTP/HTTPS (Layer 7) traffic. Moreover, when you're working with web applications, it's typical to implement routing decisions based on HTTP attributes like query parameters, headers, or cookies, a functionality that NLBs do not support due to their Layer 4 operational limitations.

The option that says: **Launch a CloudFormation stack that contains an Amazon ECS cluster that spans multiple Availability Zones using Spot Instances. Create an Application Load Balancer in front of the ECS cluster. Use the stack to launch an Amazon RDS MySQL database in Multi-AZ configuration with a "snapshot" deletion policy. Create a Route 53 zone entry for the company's domain name with an Alias-record pointed to the ALB** is incorrect. Although the Spot instance provides good cost savings for the web tier, the reliability of the site will suffer as the Spot instances are usually reclaimed by AWS based on the supply and demand of its global computing capacity. The "snapshot" deletion policy on the database tier is also not ideal as this will require a significant time to restore if you delete the CloudFormation stack.

The option that says: **Create an AWS Elastic Beanstalk application with an Auto Scaling group of EC2 instances as web servers that spans two separate regions. Put the EC2 instances behind an Application Load Balancer in each region. Launch a Multi-AZ Amazon Aurora MySQL database with cross-region read replica to the other region. Create zone entries in Route 53 with geoproximity routing policy to direct the traffic between the two regions** is incorrect. Creating two Auto Scaling groups on separate regions is unnecessary and expensive. Distributing the EC2 instance in multiple Availability Zones is enough to handle the traffic. In this setup, the database on the second region is a read-replica only, so any writes to the database will have to be sent to the main region's RDS instance.

47. QUESTION

A private bank is hosting a secure web application that allows its agents to view highly sensitive information about the clients. The amount of traffic that the web app will receive is known and not expected to fluctuate. An SSL will be used as part of the application's data security. The chief information security officer (CISO) is concerned about the security of the SSL private key. The CISO wants to ensure that the key cannot be accidentally or intentionally moved outside the corporate environment. The solutions architect is also concerned that the application logs might contain some sensitive information. The EBS volumes used to store the data are already encrypted. In this scenario, the application logs must be stored securely and durably so that they can only be decrypted by authorized employees.

Which of the following is the most suitable and highly available architecture that can meet all of the requirements?

- Distribute traffic to a set of web servers using an Elastic Load Balancer. To secure the SSL private key, upload the key to the load balancer and configure the load balancer to offload the SSL traffic. Lastly, write your application logs to an instance store volume that has been encrypted using a randomly generated AES key.
- Distribute traffic to a set of web servers using an Elastic Load Balancer that performs TCP load balancing. Use an AWS CloudHSM to perform the SSL transactions and deliver your application logs to a private Amazon S3 bucket using server-side encryption.
- (selected) Distribute traffic to a set of web servers using an Elastic Load Balancer that performs TCP load balancing. Use CloudHSM deployed to two Availability Zones to perform the SSL transactions and deliver your application logs to a private Amazon S3 bucket using server-side encryption.
- Distribute traffic to a set of web servers using an Elastic Load Balancer. Use TCP load balancing for the load balancer and configure your web servers to retrieve the SSL private key from a private Amazon S3 bucket on boot. Use another private Amazon S3 bucket to store your web server logs using Amazon S3 server-side encryption.

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use your own encryption keys on the AWS Cloud. With CloudHSM, you can manage your own encryption keys and automate time-consuming administrative tasks for you, such as hardware provisioning, software patching, high-availability, and backups.

The correct answer is the option that says: **Distribute traffic to a set of web servers using an Elastic Load Balancer that performs TCP load balancing. Use CloudHSM deployed to two Availability Zones to perform the SSL transactions and deliver your application logs to a private Amazon S3 bucket using server-side encryption.** It uses CloudHSM for performing the SSL transaction without requiring any additional way of storing or managing the SSL private key. This is the most secure way of ensuring that the key will not be moved outside of the AWS environment. Also, it uses the highly available and durable S3 service for storing the logs. Take note that this option says "server-side encryption" and not "Amazon S3-Managed Encryption Keys", which are two different things.

71. QUESTION

A startup is building a web app that lets users post photos of good deeds in their neighborhood with a 143-character caption/article. The developers decided to write the application in ReactJS, a popular javascript framework so that it would run on the broadest range of browsers, mobile phones, and tablets. The app should provide access to Amazon DynamoDB to store the caption. The initial prototype shows that there aren't large spikes in usage.

Which option provides the most cost-effective and scalable architecture for this application?

- Configure the ReactJS client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) on an EC2 instance. This will provide signed credentials to an IAM user allowing GET and PUT operations in the DynamoDB table and the S3 bucket. You serve your mobile application out of an S3 bucket enabled as a website.
- Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or any other popular social site. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in DynamoDB. Serve your web application from an NGINX server hosted on a fleet of EC2 instances, with a load balancer and auto-scaling. Add an IAM role to the EC2 instance to allow GET and PUT operations to DynamoDB tables.
- Configure the ReactJS client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) to provide signed credentials to an IAM user. This will allow GET and PUT operations to DynamoDB. Serve your web application from an NGINX server hosted in a fleet of EC2 instances that are load-balanced and auto-scaled. Your EC2 instances are configured with an IAM role that allows GET and PUT operations in DynamoDB.
- (selected) Register the web application with a Web Identity Provider such as Google, Facebook, Amazon, or from any other popular social sites and use the `AssumeRoleWithWebIdentity` API of STS to generate temporary credentials. Create an IAM role for that web provider and set up permissions for the IAM role to allow GET and PUT operations in Amazon S3 and DynamoDB. Serve your web app out of an S3 bucket enabled as a website.

Rehosting—Otherwise known as "lift-and-shift". Many early cloud projects gravitate toward net new development using cloud-native capabilities, but in a

large legacy migration scenario where the organization is looking to scale its migration quickly to meet a business case, applications can be rehosted.

Replatforming—Sometimes, this is called "lift-tinker-and-shift." Here you might make a few cloud (or other) optimizations in order to achieve some tangible benefit, but you aren't otherwise changing the core architecture of the application. You may be looking to reduce the amount of time you spend managing database instances by migrating to a database-as-a-service platform like Amazon Relational Database Service (Amazon RDS) or migrating your application to a fully managed platform like Amazon Elastic Beanstalk.

Repurchasing—Moving to a different product. Repurchasing is a move to a SaaS platform. Moving a CRM to Salesforce.com, an HR system to Workday, a CMS to Drupal, etc.

Refactoring / Re-architecting—Re-imagining how the application is architected and developed, typically using cloud-native features. This is typically driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment. For example, migrating from a monolithic architecture to a service-oriented (or server-less) architecture to boost agility.

Retire—This strategy basically means: "Get rid of." Once you've discovered everything in your environment, you might ask each functional area who owns each application and see that some of the applications are no longer used. You can save costs by retiring these applications.

Retain—Usually this means "revisit" or do nothing (for now). Maybe you aren't ready to prioritize an application that was recently upgraded or are otherwise not inclined to migrate some applications. You can retain these applications and revisit your migration strategy.

75. QUESTION

A company is planning to build its new customer relationship management (CRM) portal in AWS. The application architecture will be using a containerized microservices hosted on an Amazon ECS cluster. A Solutions Architect has been tasked to set up the architecture and comply with the AWS security best practice of granting the least privilege. The architecture should also support the use of security groups and standard network monitoring tools at the container level to comply with the company's strict IT security policies.

Which of the following provides the MOST secure configuration for the CRM portal?

- (●) Use the `awsvpc` network mode in the task definition in your Amazon ECS Cluster. Attach security groups to the ECS tasks then use IAM roles for tasks to access other resources.
- ( ) Use the `bridge` network mode in the task definition in your Amazon ECS Cluster. Attach security groups to Amazon EC2 instances then use IAM roles for EC2 instances to access other resources.
- ( ) Use AWS App Runner to run the containerized application instead to improve security and reduce operational overhead. Select VPC and security groups accordingly for deployment. Add IAM credentials to the environment variables when launching the service.
- ( ) Use the `awsvpc` network mode in the task definition in your Amazon ECS Cluster. Attach security groups to the ECS tasks then pass IAM credentials into the container at launch time to access other AWS resources.

The option that says: **Use the `bridge` network mode in the task definition in your Amazon ECS Cluster. Attach security groups to Amazon EC2 instances then use IAM roles for EC2 instances to access other resources** is incorrect because you won't be able to attach security groups to your ECS tasks using this network mode type. This will only use the Docker's built-in virtual network which runs inside each container instance. You have to use the `awsvpc` network mode instead to allow you to use security groups and network monitoring tools at a more granular level within your tasks. Moreover, if you are using the `awsvpc` network mode, you should attach the security group to the ECS task and not to the EC2 instance.

1. awsvpc network mode (modo recomendado)

Esse modo atribui **uma ENI (Elastic Network Interface)** diretamente ao **container** (na prática, à task ECS).

✓ Características:

- Cada **task** recebe:
  - seu **próprio IP privado** dentro do VPC
  - segurança via **SG (Security Group)**
  - controle tipo EC2 (VPC-native networking)
- Isolamento de rede **forte** (cada task = 1 microservidor na VPC)
- Necessário para:
  - **Fargate**
  - ALB target type = *IP*
  - Tráfego direto container ↔ rede VPC

✓ Quando usar?

- Microservices modernos
- Integração com ALB / NLB por IP
- Requisitos de segurança por SG por task
- Fargate (obrigatório)

❗ Desvantagens:

- Cada task consome uma ENI → **pode limitar escala**
- Endereços IP podem acabar no seu subnet

▨ 2. bridge network mode

É o modo **tradicional Docker**.

✓ Características:

- Containers compartilham uma **única ENI** da EC2 host
- Containers usam **NAT interno** (docker0 bridge)
- Security Group atua no **host EC2**, não no container

✓ Quando usar?

- Workloads **antigas**
- Ambientes EC2 com containers que:
    - não precisam de IP individual
    - podem compartilhar rede
- Quando quer **alta densidade de containers** por host sem gastar IPs

❗ Desvantagens:

- Não funciona com Fargate
- Não pode ser usado com ALB target type "IP"
- Menos isolamento
- Depende fortemente da rede Docker e da host EC2

🆚 Resumo fácil

Feature	**awsvpc**	**bridge**
IP por task	✓ Sim	✗ Não (1 IP por host)
Segurança por Security Group	✓ Por task	✗ Só no host
Fargate	✓ Obrigatório	✗ Não suportado
Isolamento	Alto	Médio
Escalabilidade	Limitada por IP/subnet	Melhor densidade

Load Balancer	Suporta ALB/NLB por IP	Só CLB ou ALB via instance mode

RDS "source–replica" é o novo nome (mais neutro) para o modelo tradicional **master–replica** usado em bancos relacionais gerenciados no Amazon RDS.

É simplesmente o mecanismo de **replicação assíncrona** em que: (A replicação é **assíncrona** → a replica pode estar atrasada.

Então para leituras que exigem consistência forte (read-after-write), você deve ler sempre do **source**.)

- **Source** (antes "master") → recebe escrita +leituras
- **Replica** (antes "read replica") → recebe apenas leitura

Como funciona?

1. A aplicação envia escrita → vai para o **SOURCE**
2. O SOURCE grava localmente
3. As mudanças são replicadas para as **replicas** (geralmente assíncrono)
4. As replicas ficam defasadas alguns milissegundos (replication lag)

📌 Em quais engines isso existe no RDS?

- MySQL
- MariaDB
- PostgreSQL
- SQL Server (via outros mecanismos)

Aurora tem outro modelo (cluster com writer e readers).

57. QUESTION

A company has just launched a new central employee registry application that contains all of the public employee registration information of each staff of the company. The application has a microservices architecture running in Docker in a single AWS Region. The management teams from other departments who have their servers located in different VPCs need to connect to the central repository application to continue their work. The Solutions Architect must ensure that the traffic to the application does not traverse the public Internet. The IT Security team must also be notified of any denied requests and be able to view the corresponding source IP.

How will the Architect implement the architecture of the new application given these circumstances?

- Set up a Transit VPC by using third-party marketplace VPN appliances running on an On-Demand Amazon EC2 instance that dynamically routes the VPN connections to the virtual private gateways (VGWs) attached to each VPC. Set up an AWS Config rule on each VPC to capture rejected traffic requests, including the source IPs, that will be delivered to an Amazon CloudWatch Logs group. Set up a CloudWatch Logs subscription that streams the log data to the IT Security account.
- Set up an IPSec Tunnel between the central VPC and each of the teams' VPCs. Create VPC Flow Logs on each VPC to capture rejected traffic requests, including the source IPs, that will be delivered to an Amazon CloudWatch Logs group. Create a CloudWatch Logs subscription that streams the log data to the IT Security account.
- (●) Link each of the teams' VPCs to the central VPC using VPC Peering. Create VPC Flow Logs on each VPC to capture rejected traffic requests, including the source IPs, that will be delivered to an Amazon CloudWatch Logs group. Set up a CloudWatch Logs subscription that streams the log data to the IT Security account.
- Use AWS Direct Connect to create a dedicated connection between the central VPC and each of the teams' VPCs. Enable the VPC Flow Logs on each VPC to capture rejected traffic requests, including the source IPs, that will be delivered to a CloudWatch Logs group. Set up an Amazon CloudWatch Logs subscription that streams the log data to the IT Security account.

The option that says: **Set up an IPSec Tunnel between the central VPC and each of the teams' VPCs. Create VPC Flow Logs on each VPC to capture rejected traffic requests, including the source IPs, that will be delivered to an Amazon CloudWatch Logs group. Create a CloudWatch Logs subscription that streams the log data to the IT Security account** is incorrect. It is mentiond in the scenario that the traffic to the application must not traverse the public Internet. Since an IPSec tunnel uses the Internet to transfer data from your VPC to a specified destination, this solution is definitely incorrect.

Um **IPSec Tunnel** é como um "**canal secreto**" entre duas redes, onde tudo que passa por dentro é **criptografado**, impedindo espionagem ou alteração. Ele funciona encapsulando e criptografando pacotes IP, permitindo que redes privadas se comuniquem através da **internet pública de forma segura**

Amazon GuardDuty é um **serviço de detecção de ameaças** totalmente gerenciado pela AWS.
Ele monitora continuamente sua conta, workloads e dados para identificar **comportamentos suspeitos, ataques, acessos anormais e possíveis violações de segurança**.

☑ O que o GuardDuty faz?

Ele analisa automaticamente vários sinais de segurança, como:

- Logs do **CloudTrail**
- Logs de **VPC Flow Logs**
- Logs do **DNS**
- Atividade de **EKS (Kubernetes)**

- Insights de inteligência de ameaças (AWS + parceiros)

E usa machine learning + detecção de anomalias para identificar riscos.

GuardDuty detecta:

🔓 Acesso não autorizado

- Login suspeito vindo de outro país
- Uso de chaves da AWS comprometidas
- Atividade incomum em IAM, EC2, S3

🥷 Movimentação lateral

- Comunicação estranha entre instâncias
- Portas incomuns sendo acessadas
- Tentativas de escalonamento de privilégios

🐖 Instâncias comprometidas

- EC2 se conectando a botnets
- Mineração de criptomoedas
- Malware comunicando com servidores C2

📦 Ameaças em dados

- Acesso incomum a buckets S3
- Exfiltração de dados
- Downloads anômalos

☸️ EKS / Containers

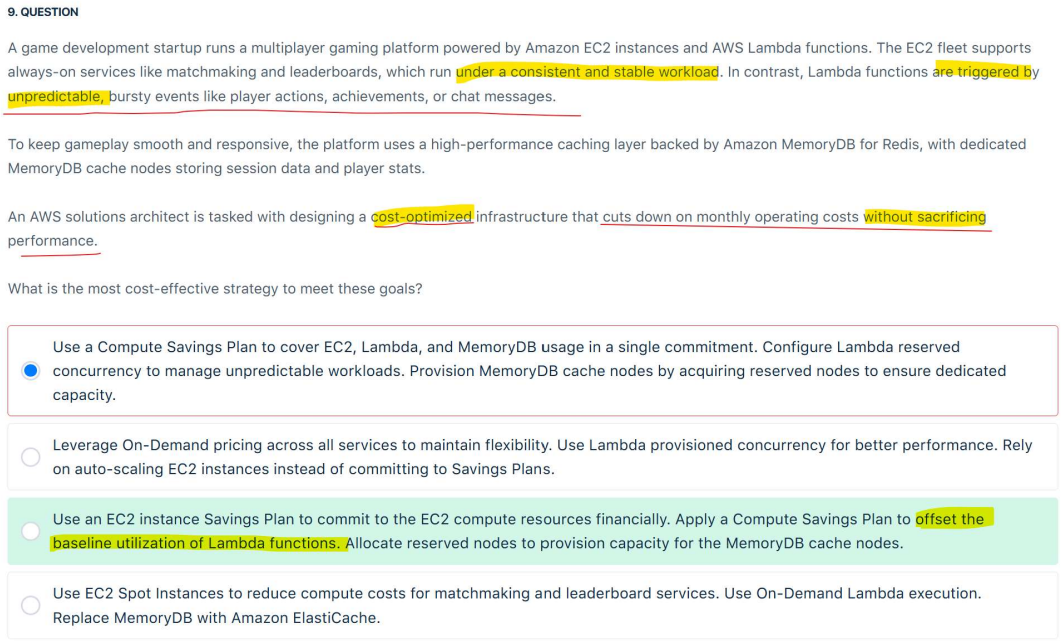
- Execução de comandos suspeitos no cluster
- Imagens maliciosas
- Pods conversando com IPs maliciosos

🤯 Como ele funciona?

1. **Habilita com 1 clique**

   Não precisa instalar agentes.

2. GuardDuty começa a analisar continuamente logs e eventos.
3. Identifica comportamentos suspeitos e gera **Findings** (alertas).
4. Você visualiza os findings no console e pode integrar com:
   - EventBridge → para automatizar ações
   - Security Hub → centralizar alertas
   - Lambda → remediação automática
   - SIEM / SOAR



9. QUESTION

A game development startup runs a multiplayer gaming platform powered by Amazon EC2 instances and AWS Lambda functions. The EC2 fleet supports always-on services like matchmaking and leaderboards, which run under a consistent and stable workload. In contrast, Lambda functions are triggered by unpredictable, bursty events like player actions, achievements, or chat messages.

To keep gameplay smooth and responsive, the platform uses a high-performance caching layer backed by Amazon MemoryDB for Redis, with dedicated MemoryDB cache nodes storing session data and player stats.

An AWS solutions architect is tasked with designing a cost-optimized infrastructure that cuts down on monthly operating costs without sacrificing performance.

What is the most cost-effective strategy to meet these goals?

- (●) Use a Compute Savings Plan to cover EC2, Lambda, and MemoryDB usage in a single commitment. Configure Lambda reserved concurrency to manage unpredictable workloads. Provision MemoryDB cache nodes by acquiring reserved nodes to ensure dedicated capacity.
- ( ) Leverage On-Demand pricing across all services to maintain flexibility. Use Lambda provisioned concurrency for better performance. Rely on auto-scaling EC2 instances instead of committing to Savings Plans.
- ( ) Use an EC2 instance Savings Plan to commit to the EC2 compute resources financially. Apply a Compute Savings Plan to offset the baseline utilization of Lambda functions. Allocate reserved nodes to provision capacity for the MemoryDB cache nodes.
- ( ) Use EC2 Spot Instances to reduce compute costs for matchmaking and leaderboard services. Use On-Demand Lambda execution. Replace MemoryDB with Amazon ElastiCache.

The Lambda functions, which respond to bursty and inconsistent traffic patterns, should be covered using a Compute Savings Plan, but only up to the minimum baseline usage that can be forecasted. This configuration still allows cost savings without overcommitting.Lambda 函数用于应对突发性和不稳定的流量模式，应使用计算资源节省计划进行覆盖，但仅限于可预测的最低基准使用量。这种配置既能节省成本，又不会造成资源过度投入。

5. QUESTION

A global enterprise manages multiple AWS accounts under a single organization using AWS Organizations. The enterprise leverages AWS CloudFormation to automate infrastructure provisioning and centrally manages permissions and roles using AWS Identity and Access Management (IAM).

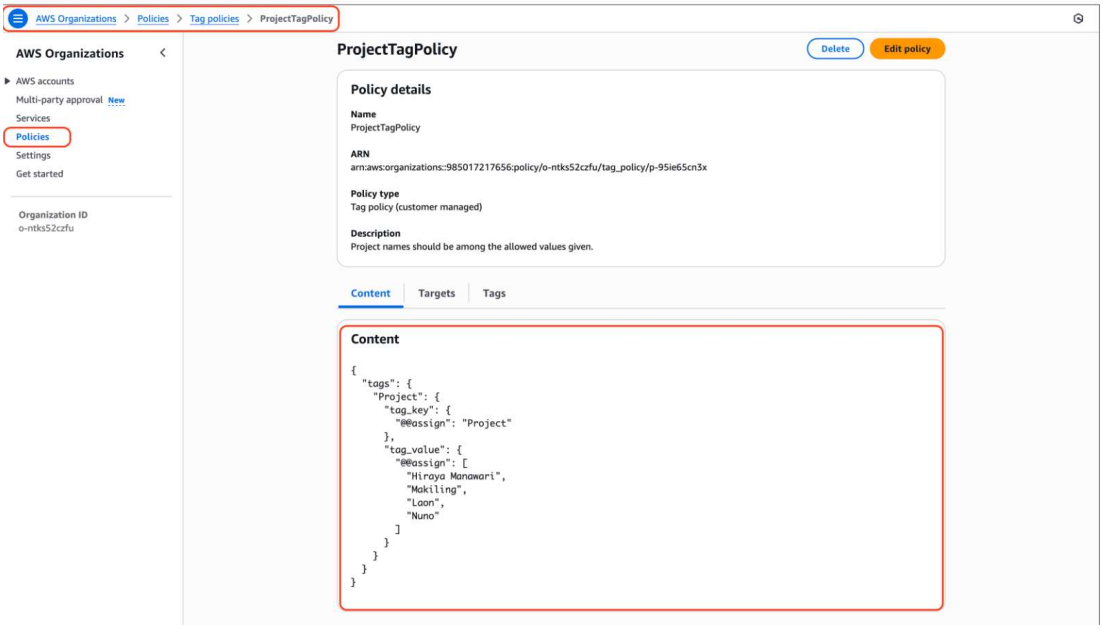
As part of a new chargeback model, the finance department requires each department to apply project tags selected from a predefined list of tag values to every newly created resource.

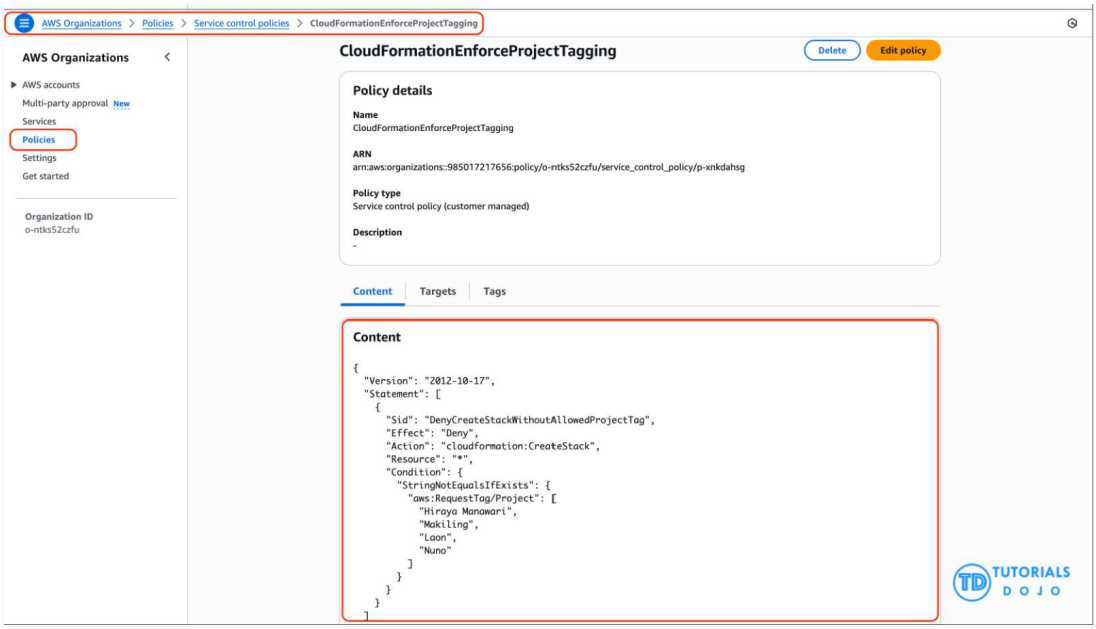
When analyzing spending using the AWS Cost and Usage Report in AWS Cost Explorer, the finance team identified non-compliant tag values that disrupted cost attribution.

To enforce compliance, the company needs a scalable way to validate and implement resource tagging based on these project values.

Which approach offers the easiest and most effective way to enforce this?

- (●) In the organization's management account, define a tag policy with valid project tag values. Then, create a Service Control Policy (SCP) to deny `cloudformation:CreateStack` except if the required project tag is included. Attach the created SCP to the root of the organization.
- ( ) Configure a tag policy in the Organizations management account to specify the approved project tag values. Then, implement an SCP that blocks the `cloudformation:DetectStackResourceDrift` operation unless a compliant project tag is part of the request. Apply this SCP at the organization's root level.
- ( ) Enable resource tagging enforcement using AWS Config rules. Integrate the rules with Organizations to evaluate tag compliance across accounts. Use automation documents to prevent `cloudformation:CreateStack` unless tags meet policy requirements.
- ( ) Use the AWS Tag Editor to audit and correct resource tags across all accounts. Schedule periodic reviews to identify non-compliant tags and update them to match the predefined project tag list





The option that says: **Configure a tag policy in the Organizations management account to specify the approved project tag values. Then, implement an SCP that blocks the `cloudformation:DetectStackResourceDrift` operation unless a compliant project tag is part of the request. Apply this SCP at the organization's root level** is incorrect. It targets an API operation unrelated to resource creation where tag enforcement must occur. While defining a tag policy in the AWS Organizations management account helps establish governance, denying `cloudformation:DetectStackResourceDrift` is ineffective. This action is used for diagnostics and does not create, modify, or tag resources. Blocking it does nothing to prevent the deployment of resources with invalid or missing tags.

7. QUESTION

A bank has received a security directive to harden its cloud infrastructure in response to a wave of cyberattacks targeting financial institutions. As part of this initiative, the bank is integrating third-party firewall appliances sourced from the AWS Marketplace to perform deep inspection and control of outbound internet traffic originating from all internal workloads.

The strategy involves centralizing these firewall appliances inside a shared services VPC, serving as a security chokepoint across the organization's cloud infrastructure. The bank's network team has already implemented VPC routing that funnels traffic from spoke VPCs into the shared services VPC.

The cloud security engineer must design a secure and highly available inspection architecture within a single AWS Region to satisfy security requirements and uptime guarantees.

Which of the following should the cloud engineer implement to meet these goals?

- Provision firewall appliances across two different Availability Zones in the shared services VPC. Set up a Network Load Balancer linked to a target group that includes the appliances. Deploy a VPC interface endpoint and configure the route table to direct incoming VPC traffic to this endpoint as the next hop.
- Launch both firewall appliances in one Availability Zone within the shared services VPC. Configure a Gateway Load Balancer and associate it with a target group containing the appliances. Provision a Gateway Load Balancer endpoint and update the shared services VPC route table to use it as the next hop for incoming traffic from connected VPCs.
- Launch both firewall appliances in a single Availability Zone within the shared services VPC. Configure an Application Load Balancer and link it to a target group containing the appliances. Deploy a VPC interface endpoint and update the route table to forward traffic from connected VPCs to the endpoint as the next hop.
- Place two firewall appliances in different Availability Zones in the shared services VPC. Set up a Gateway Load Balancer and link it to a target group containing both appliances. Then, deploy a Gateway Load Balancer endpoint and configure the shared services VPC route table to forward inter-VPC traffic to this endpoint as the next hop.

Launch both firewall appliances in one Availability Zone within the shared services VPC. Configure a Gateway Load Balancer and associate it with a target group containing the appliances. Provision a Gateway Load Balancer endpoint and update the shared services VPC route table to use it as the

next hop for incoming traffic from connected VPCs is incorrect. This option proposes correctly using the Gateway Load Balancer and the GWLB endpoint. Still, it fails the high availability requirement because it places both firewall appliances in the **same** Availability Zone. This setup introduces **a single point of failure,** which the scenario explicitly aims to avoid. Even though the components are set up correctly, the lack of cross-AZ deployment makes it non-compliant with uptime and regulatory demands. Just distributing appliances across zones would have made it a stronger candidate.

10. QUESTION

A fast-rising tech company is preparing to shift a portfolio of applications to AWS. The current on-premises setup includes a mix of physical and virtual machines. The founder CTO, who started the IT infrastructure architecture, left the company and provided scant documentation on the architecture. The company lacked complete visibility into its application estate and infrastructure layout.

One of the critical applications slated for migration has several unmapped dependencies, with many using low-latency communications over a custom IP-based protocol bound to port 1000. To preserve operational integrity, the company aims to move this application and all tightly-coupled dependencies simultaneously.

The company has deployed the AWS Application Discovery Agent across its servers and collected telemetry for several months to gain insights. Now, the company must identify the complete set of connected systems that must be migrated together in the same move group.

What approach should the company's new CTO take to discover and group the correct dependencies for simultaneous migration?

- ○ Use AWS Application Migration Service to replicate the application servers, then migrate them immediately. Rely on Amazon CloudWatch metrics after migration to identify any issues retroactively. Once the application is running in AWS, use the AWS Application Discovery Agent to scan the new environment for missing dependencies and migrate those systems later as needed.
- ○ Export the AWS Application Discovery Agent data to Amazon S3. Use AWS Glue to crawl the data and create a data catalog. Then run Amazon Athena queries to search for references to port 1000 or IP traffic. Create a move group based on the query results and proceed with the migration.
- ● Leverage AWS Migration Hub to select the application servers. Use the network graph to detect communication patterns. Enable data exploration via Amazon Athena and analyze traffic logs for connections on port 1000. Identify all dependent servers involved in low-latency communications using the custom IP protocol. Use this analysis to define a move group in Migration Hub for coordinated migration.
- ○ Use the AWS Network Access Analyzer console to define a scope that matches traffic on port 1000. Analyze traffic flows to determine which systems are communicating with the application. Then, create a move group in AWS Migration Hub based on the results of the Network Access Scope.

11. QUESTION

A cloud engineer at a fintech startup is preparing the infrastructure for a critical internal platform launch. The cloud engineer configures DNS resolution for a newly provisioned VPC in the ap-southeast-1 (Singapore) Region as part of the setup. This VPC uses the 13.232.18.0/24 CIDR block and has been integrated with Amazon Route 53 Resolver to manage DNS queries.

Initially, the team planned for basic DNS resolution, but the requirements have since changed to support more advanced functionality. Now, all domain name lookups must be routed through a Private hosted zone, and any Amazon EC2 instances with assigned Public IP addresses must automatically receive corresponding Public hostnames.

Which of the following should the cloud engineer do to satisfy the new requirements and ensure the domain names are resolved correctly and consistently within the VPC? (Select TWO.)

- ☑ Set up a private DNS zone and link it to the VPC. Confirm that the VPC has both `enableDnsSupport` and `enableDnsHostnames` enabled.
- ☐ Create a private hosted zone and associate it with the VPC. Ensure the VPC is configured with the appropriate `instanceTenancy` setting.
- ☐ Create a new DHCP options set that sets `domain-name-servers=13.232.18.98` , and apply this configuration to the VPC by associating it accordingly.
- ☑ Define a custom DHCP options set that specifies `domain-name-servers=AmazonProvidedDNS` , and attach this configuration to the VPC.
- ☐ Establish a private hosted zone and connect it to the designated VPC. Make sure that the `enableNetworkAddressUsageMetrics` feature is activated.

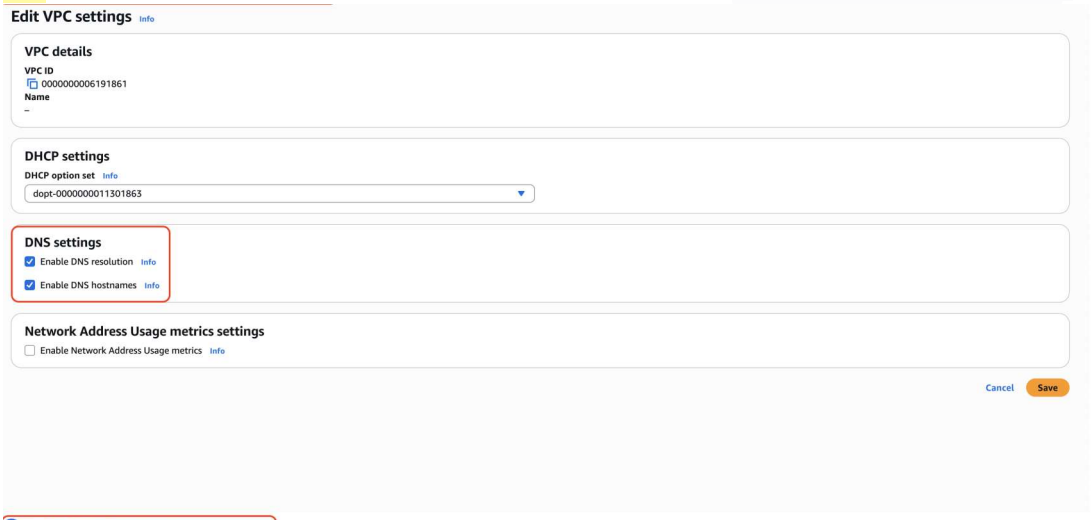
enabling `enableDnsSupport` allows instances in the VPC to use a DNS resolver.

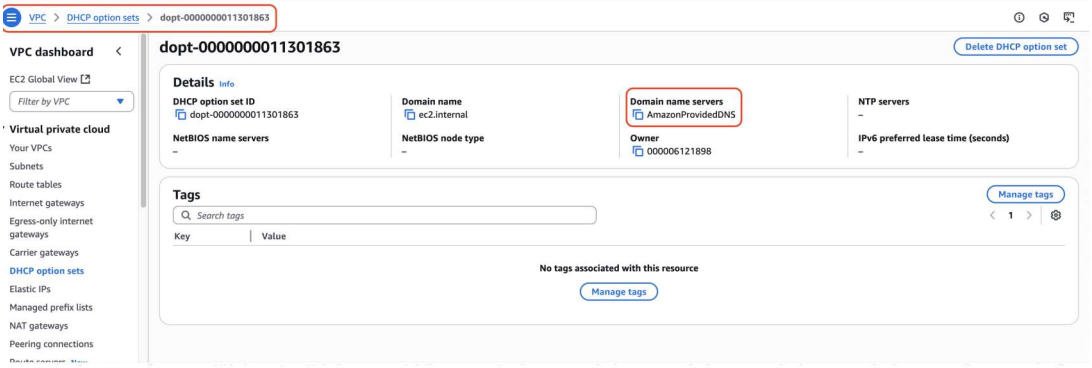
Enabling `enableDnsHostnames` ensures that instances assigned **public IP addresses receive corresponding public DNS hostnames**.

Defining a custom DHCP options set that specifies `domain-name-servers=AmazonProvidedDNS` ensures compatibility with Amazon Route 53 Resolver, including support for **private hosted zones and automatic assignment of public hostnames** to EC2 instances with **public IPs.**

private hosted zones and automatic assignment of public hostnames to EC2 instances with **public IPs.**

-》 2 个设置 -》 1 DNS（2 SETTINGS enable) + DHCP (`AmazonProvidedDNS`)





14. QUESTION

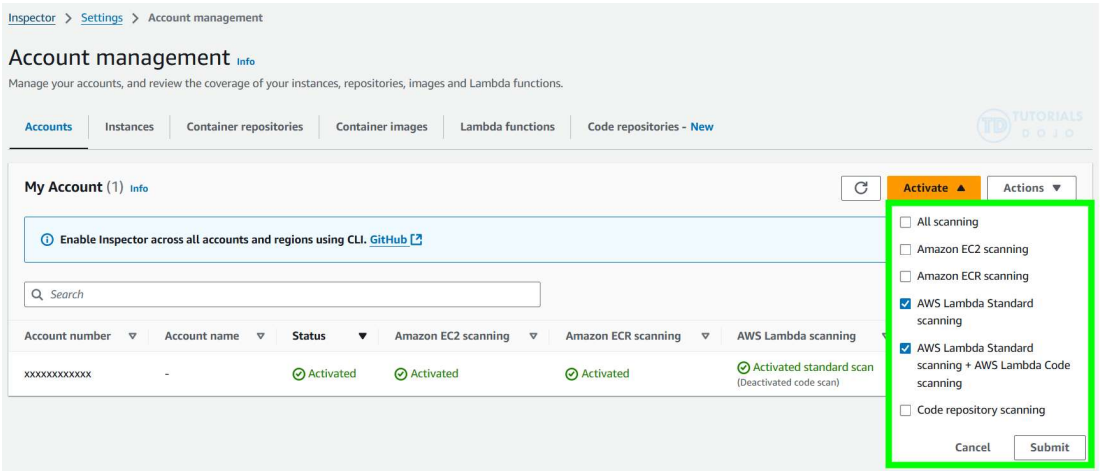
A financial institution is running multiple AWS Lambda functions. These functions are packaged as .zip archives and rely on a shared Lambda layer that contains common libraries.

Security compliance requires that the development team must:

- Continuously monitor the Lambda functions and the shared Lambda layer for current common vulnerabilities and exposures (CVEs).
- Perform additional automated code scans on a subset of Lambda functions to detect data leaks and injection flaws.
- Ensure that not all Lambda functions are scanned for code issues, since some workloads handle non-sensitive test data.
- The company wants a fully managed AWS-native solution to meet these requirements.

Which of the following will meet the given requirements? (Select THREE.)

- [ ] Use AWS IAM policies to restrict Amazon Inspector from scanning specific Lambda functions.
- [x] Enable Amazon Inspector Lambda code scanning to detect code-level vulnerabilities.
- [x] Enable Amazon Inspector and activate Lambda standard scanning for CVE detection.
- [x] Apply an Amazon Inspector exclusion tag with a key of `InspectorCodeExclusion` and a value of `LambdaCodeScanning` to Lambda functions that should not be code-scanned.
- [ ] Use Amazon GuardDuty Lambda Protection to monitor for CVEs and data leaks.
- [ ] Enable AWS Config with a custom rule to check Lambda deployment packages for CVEs.



15. QUESTION

An international shipping company with a global fleet runs its cloud operations through a multi-account AWS environment managed by AWS Control Tower. Due to fluctuating fuel costs and rising logistics expenses, the company recently launched a company-wide cost-cutting initiative, including the tech department.

The company uses AWS Organizations for account-level boundaries, AWS Config to ensure compliance standards are met, and AWS Trusted Advisor to surface cost inefficiencies and potential security gaps.

As per the directive of the CTO, in line with company cost-cutting, only Amazon EC2 and Amazon RDS burstable instances should be allowed. All other services not directly supporting the company's lean tech initiative must be blocked.

A solutions architect has been tasked with locking this down in a scalable, maintainable way aligned with AWS best practices.

What's the most effective approach to enforce these service restrictions?

- (●) Define a custom preventive guardrail within AWS Control Tower that enforces restrictions allowing only burstable instance types to be launched, while blocking access to any non-essential services. Assign this control to the target development OU to ensure policy compliance across applicable accounts.
- ( ) Set up a custom detective guardrail in AWS Control Tower to monitor the environment for compliance, ensuring that only burstable instances are provisioned and that any usage of unauthorized services is flagged. Link this control to the corresponding OU to track adherence across associated accounts.
- ( ) Deploy Amazon GuardDuty to scan for unsupported AWS services and non-burstable EC2 or RDS instances. Use its findings to block non-compliant deployments automatically in the future.
- ( ) Enable AWS Trusted Advisor's cost optimization checks in each development account to enforce the usage of burstable instances and prevent launching non-approved services.

16. QUESTION

A tech company that performs large-scale analytics uses an Amazon Redshift cluster with reserved nodes to support its daily data workloads. A recently hired director has initiated a new requirement: generating a deep audit analysis report. This new reporting initiative has introduced complex read queries that are highly CPU-intensive, resulting in unexpected bursts of usage on the Redshift cluster.

The existing setup must continue to handle both read and write queries consistently, even during performance spikes. A solutions architect must propose the most cost-effective solution to address the rising CPU metrics while maintaining system responsiveness and minimizing operational cost.

What is the most economical approach to manage the increased workload introduced by this new reporting initiative?

- ○ Export data from Redshift and import it into Amazon RDS PostgreSQL for deep audit analysis during periods of high load.
- ○ Launch additional Amazon EC2 instances to scale horizontally Redshift for query parallelization.
- ● Enable concurrency scaling in the Redshift setup to handle spikes in concurrent query loads without impacting performance.
- ○ Deploy an Amazon EMR cluster to offload high-load transformation jobs from the main analytics pipeline.

1. Redshift RA3 Clusters – Concurrency Scaling (auto-scaling de capacidade)

Você tem um cluster principal (RA3).
Quando muitas queries chegam ao mesmo tempo, o Redshift ativa Concurrency Scaling clusters.
Eles aumentam capacidade de processamento automaticamente.
Quando o pico acaba, os clusters extras são desligados.
👉 É auto-scaling horizontal para lidar com concorrência.

2. Redshift Serverless – Auto-scaling TOTAL

Se você usa Redshift Serverless, aí sim o autoscaling é completo:
Funciona assim:
Não existe cluster fixo.
O Redshift aumenta ou reduz automaticamente:
memória
CPU
I/O
Você só paga pelo uso (RPU — Redshift Processing Units).
👉 É a forma mais simples de auto-scaling do Redshift.