Question 1: **Correct**

You have set up a security group for your bastion host that only allows SSH from your IP:

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP 109.190.217.138/32
SSH	TCP	22	Custom sg-1a56897f

Yet when looking at the VPC Flow Logs with AWS Athena, you see a lot of instances with the IP starting with `172.XXX.XXX.XXX` also being able to issue SSH commands.

Why is that so?

- ◯ **Someone is attacking your EC2 instance, use AWS Inspector to verify that**
- ◯ **The IP rule should be `109.190.217.138/0`**
- ◯ **The second rule allows EC2 instances from an entire security group to SSH into your bastion host** **(Correct)**
- ◯ **The NACL rules are too open**

Explanation

Correct option:

The second rule allows EC2 instances from an entire security group to SSH into your bastion host

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

The following are the characteristics of security group rules:

By default, security groups allow all outbound traffic.

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

Security groups are stateful

As all the rules in the Security Group are aggregated together, therefore, the second rule allows all the instances in the mentioned Security Group to SSH into the bastion host.

Incorrect options:

The IP rule should be `109.190.217.138/0` - A CIDR block of /0 would allow access to any IP address between 0.0.0.0 and 255.255.255.255, while a CIDR block of /32 would only allow access to the IP address that precedes it. Therefore this option is not correct.

The NACL rules are too open - A Network Access Control List (NACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups to add an additional layer of security to your VPC.

This option is a distractor. Even if the NACL rules are too open, the Security Group can be used to allow only the desired traffic.

Someone is attacking your EC2 instance, use AWS Inspector to verify that - Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. This option has been added as a distractor.

Reference:

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

Question 2: **Correct**

How should MFA-Delete be enabled on an S3 bucket?

- ○ **Using the root account and the AWS Console**
- ○ **Using the root account and the AWS CLI** **(Correct)**
- ○ **Using an admin IAM user and the AWS CLI**
- ○ **Using an admin IAM user and the AWS Console**

Explanation

Correct option:

Using the root account and the AWS CLI

MFA Delete represents another layer of security wherein you can configure a bucket to enable MFA (multi-factor authentication) Delete, which requires additional authentication for either of the following operations:

Change the versioning state of your bucket

Permanently delete an object version

You should note that only the bucket owner (root account) can enable MFA Delete only via the AWS CLI. However, the bucket owner, the AWS account that created the bucket (root account), and all authorized IAM users can

enable versioning.

MFA delete

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

- Change the versioning state of your bucket
- Permanently delete an object version

MFA Delete requires two forms of authentication together:

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

MFA Delete thus provides added security in the event, for example, your security credentials are compromised.

To enable or disable MFA Delete, you use the same API that you use to configure versioning on a bucket. Amazon S3 stores the MFA Delete configuration in the same *versioning* subresource that stores the bucket's versioning status.

MFA Delete can help prevent accidental bucket deletions by doing the following:

- Requiring the user who initiates the delete action to prove physical possession of an MFA device with an MFA code.
- Adding an extra layer of friction and security to the delete action.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Note

The bucket owner, the AWS account that created the bucket (root account), and all authorized IAM users can enable versioning, but only the bucket owner (root account) can enable MFA Delete. For more information, see the AWS blog post on MFA Delete and Versioning.

To use MFA Delete, you can use either a hardware or virtual MFA device to generate an authentication code. The following example shows a generated authentication code displayed on a hardware device.

via - https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html#MultiFactorAuthenticationDelete

Incorrect options:

Using the root account and the AWS Console

Using an admin IAM user and the AWS Console

Using an admin IAM user and the AWS CLI

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

Reference:

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

Question 3: **Correct**

As part of your yearly compliance report, it has been noted that many of your EC2 instances have been lagging with their OS patches updates. You have decided to use SSM to patch these instances regularly. To meet regulatory guidelines, you need to provide a report showing that no outstanding known vulnerabilities are left unpatched. This report must be generated weekly.

Which service should you use?

- ○ **AWS Shield**
- ○ **AWS GuardDuty**
- ○ **AWS Inspector** **(Correct)**
- ○ **AWS SSM**

Explanation

Correct option:

AWS Inspector

Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances.

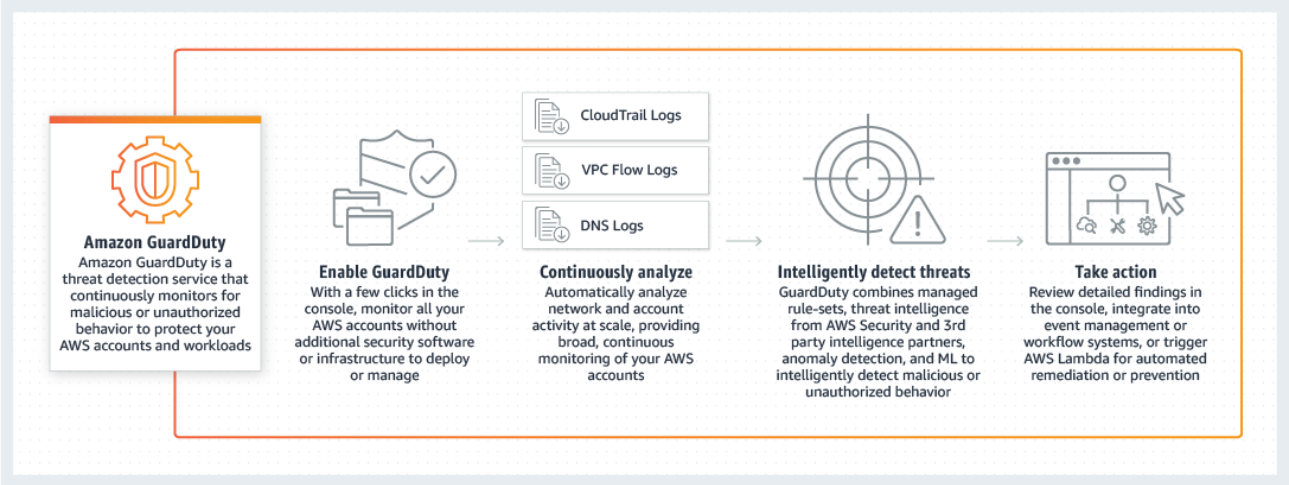
An Amazon Inspector assessment report can be generated for an assessment run once it has been successfully completed. An assessment report is a document that details what is tested in the assessment run, and the results of the assessment. The results of your assessment are formatted into a standard report, which can be generated to share results within your team for remediation actions, to enrich compliance audit data, or to store for future reference.

You can select from two types of report for your assessment, a findings report or a full report. The findings report contains an executive summary of the assessment, the instances targeted, the rules packages tested, the rules that generated findings, and detailed information about each of these rules along with the list of instances that failed the check. The full report contains all the information in the findings report and additionally provides the list of rules that were checked and passed on all instances in the assessment target.

Incorrect options:

AWS GuardDuty - GuardDuty is a threat detection service that monitors malicious activity and unauthorized behavior to protect your AWS account. GuardDuty analyzes billions of events across your AWS accounts from AWS CloudTrail (AWS user and API activity in your accounts), Amazon VPC Flow Logs (network traffic data), and DNS Logs (name query patterns). GuardDuty cannot provide a report showing that no outstanding known vulnerabilities are left unpatched.

How GuardDuty Works:



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

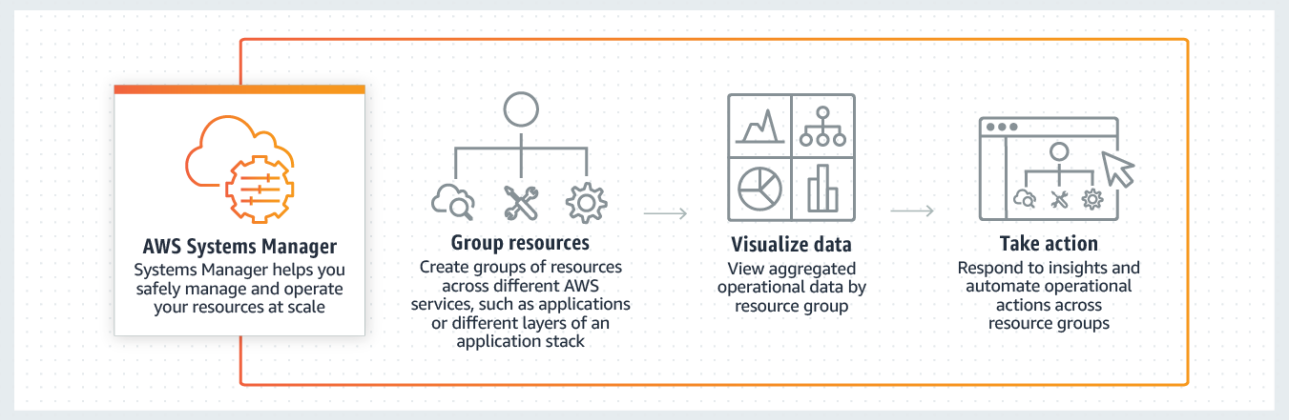
AWS Shield - AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. AWS Shield cannot provide a report showing that no outstanding known vulnerabilities are left unpatched.

"AWS SSM" - AWS Systems Manager (SSM) gives you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks such as running commands, managing patches, and configuring servers across AWS Cloud as well as on-premises infrastructure.

With Systems Manager, you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources.

Systems Manager cannot provide a report showing that no outstanding known vulnerabilities are left unpatched.

How Systems Manager Works:



via - https://aws.amazon.com/systems-manager/

References:

https://aws.amazon.com/inspector/

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

https://aws.amazon.com/systems-manager/

Question 4: **Correct**

You would like to replace your on-premise NFS v3 drive with something that will leverage the huge capacity of Amazon S3. You would like to ensure files that are commonly used are locally cached on-premises.

What should you use?

EBS Drives

Volume Gateway

File Gateway (Correct)

EFS

Explanation

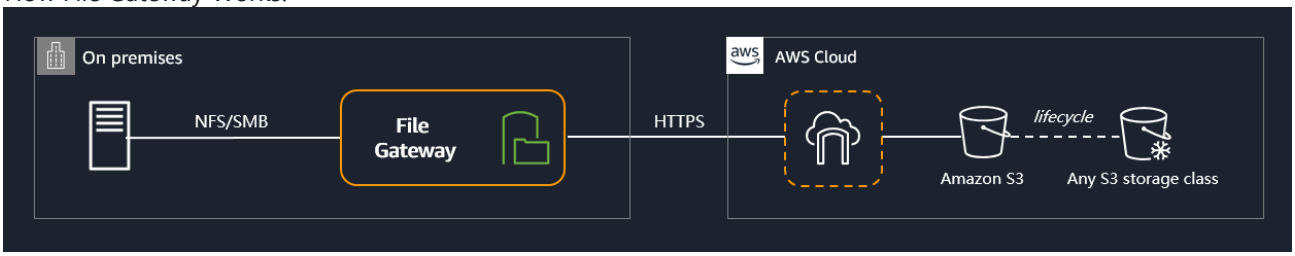
Correct option:

File Gateway - File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises applications, and for Amazon EC2-based applications that need file protocol access to S3 object storage.

How Storage Gateway Works:

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

How File Gateway Works:



via - https://aws.amazon.com/storagegateway/file/

Incorrect options:

EFS - Amazon EFS is a file storage service for use with Amazon EC2. Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for up to thousands of Amazon EC2 instances. Amazon S3 is an object storage service. EFS cannot leverage S3 for storage.

EBS Drives - Amazon Elastic Block Store (EBS) is an easy to use, high-performance, block-storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS. EBS cannot leverage S3 for storage.

Volume Gateway - You can configure the AWS Storage Gateway service as a Volume Gateway to present cloud-based iSCSI block storage volumes to your on-premises applications. The Volume Gateway provides either a local cache or full volumes on-premises while also storing full copies of your volumes in the AWS cloud. Volume Gateway also provides Amazon EBS Snapshots of your data for backup, disaster recovery, and migration. It's easy to get started with the Volume Gateway: Deploy it as a virtual machine or hardware appliance, give it local disk resources, connect it to your applications, and start using your hybrid cloud storage for block data. Since Volume Gateway represents cloud-backed iSCSI block storage, so it cannot be used to replace an on-premise NFS v3 drive, so this option is incorrect.

How Volume Gateway Works:

via - https://aws.amazon.com/storagegateway/volume/

References:

https://aws.amazon.com/storagegateway/file/

https://aws.amazon.com/storagegateway/volume/

https://aws.amazon.com/storagegateway/

Question 5: **Correct**

You just released a new mobile game and users have the chance to interact with each other. In order to publish a profile picture, your company has made the architectural decision to have users directly upload their images into a designated S3 bucket.

How can you provide write access to the mobile application users effectively?

- ○ **Create one IAM user and publish the access keys as part of the mobile application**

- ○ **Federate the users with SAML so they can use Single Sign-On (SSO) to access S3**

○ **Federate the users with Cognito so they can assume a role to access S3** **(Correct)**

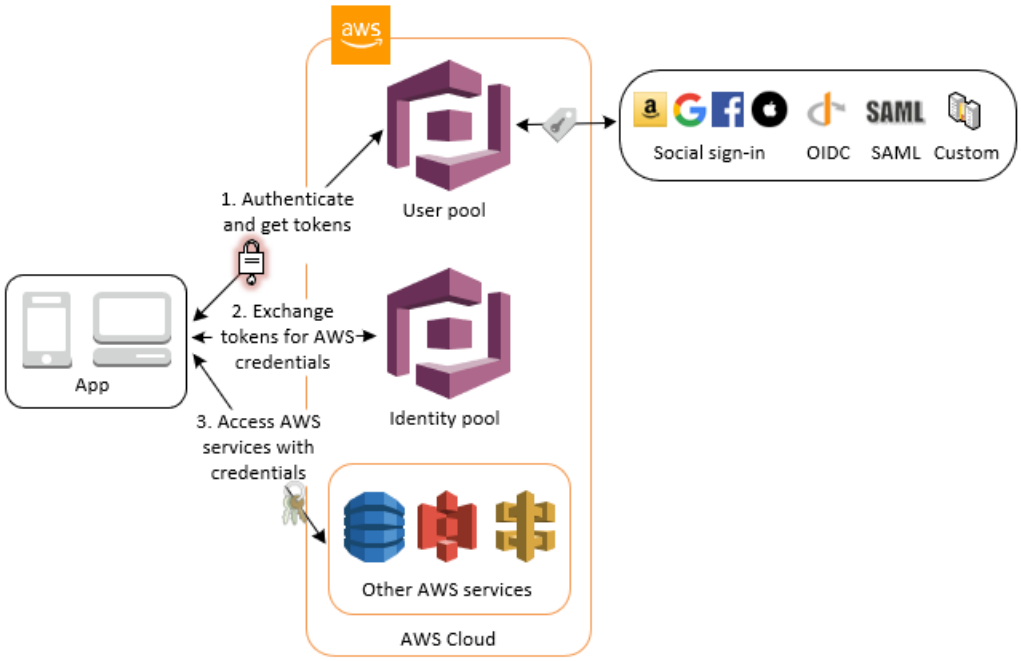
○ **Create an AWS Lambda function that will create an IAM User for each new user, and store their API keys in the mobile app database**

Explanation

Correct option:

Federate the users with Cognito so they can assume a role to access S3

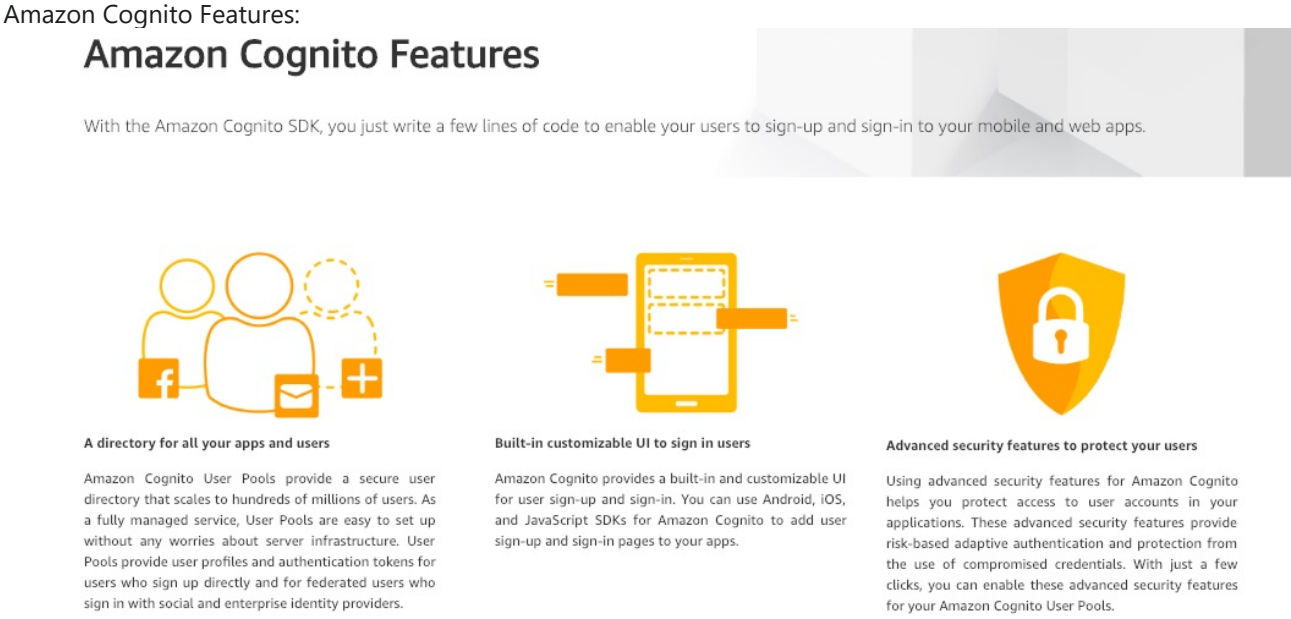
Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0. Application-specific user authentication can be provided via a Cognito User Pool and then users can access AWS services such as S3 using a Cognito Identity Pool. Here Cognito is the best technology choice for managing mobile user accounts.



via

- https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html

Amazon Cognito Features:



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

Exam Alert:

Please review the following note to understand the differences between Cognito User Pools and Cognito Identity Pools:

Features of Amazon Cognito

User pools

A user pool is a user directory in Amazon Cognito. With a user pool, your users can sign in to your web or mobile app through Amazon Cognito, or federate through a third-party identity provider (IdP). Whether your users sign in directly or through a third party, all members of the user pool have a directory profile that you can access through an SDK.

User pools provide:

- Sign-up and sign-in services.
- A built-in, customizable web UI to sign in users.
- Social sign-in with Facebook, Google, Login with Amazon, and Sign in with Apple, and through SAML and OIDC identity providers from your user pool.
- User directory management and user profiles.
- Security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.
- Customized workflows and user migration through AWS Lambda triggers.

For more information about user pools, see Getting Started with User Pools and the Amazon Cognito User Pools API Reference.

Identity pools

With an identity pool, your users can obtain temporary AWS credentials to access AWS services, such as Amazon S3 and DynamoDB. Identity pools support anonymous guest users, as well as the following identity providers that you can use to authenticate users for identity pools:

- Amazon Cognito user pools
- Social sign-in with Facebook, Google, Login with Amazon, and Sign in with Apple
- OpenID Connect (OIDC) providers
- SAML identity providers
- Developer authenticated identities

To save user profile information, your identity pool needs to be integrated with a user pool.

For more information about identity pools, see Getting Started with Amazon Cognito Identity Pools (Federated Identities) and the Amazon Cognito Identity Pools API Reference.

via - https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html

Incorrect options:

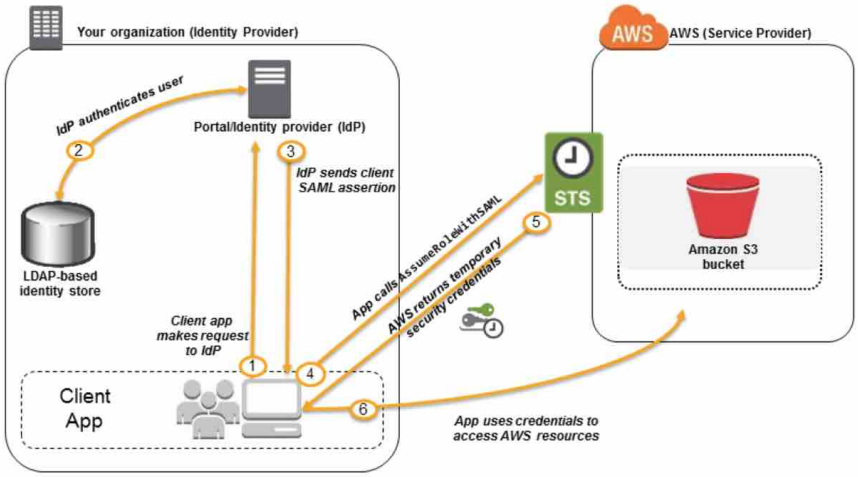
Create an AWS Lambda function that will create an IAM User for each new user, and store their API keys in the mobile app database - Creating an IAM user for each new user of the mobile phone is not practicable, so this option is ruled out.

Create one IAM user and publish the access keys as part of the mobile application - This is a security bad-practice. You should not expose the IAM user access keys via a third-party application. The best solution is to use Cognito user pool for authentication and then access AWS services using an identity pool.

Federate the users with SAML so they can use Single Sign-On (SSO) to access S3 - The scenario does not mention that the users belong to a specific organization, therefore you cannot use SAML to facilitate SSO to access S3.

Using SAML-based federation for API access to AWS

Assume that you want to provide a way for employees to copy data from their computers to a backup folder. You build an application that users can run on their computers. On the back end, the application reads and writes objects in an S3 bucket. Users don't have direct access to AWS. Instead, the following process is used:



1. A user in your organization uses a client app to request authentication from your organization's IdP.
2. The IdP authenticates the user against your organization's identity store.
3. The IdP constructs a SAML assertion with information about the user and sends the assertion to the client app.
4. The client app calls the AWS STS `AssumeRoleWithSAML` API, passing the ARN of the SAML provider, the ARN of the role to assume, and the SAML assertion from IdP.
5. The API response to the client app includes temporary security credentials.
6. The client app uses the temporary security credentials to call Amazon S3 API operations.

via - https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

References:

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

https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html

https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools.html

https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html

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

Question 6: **Correct**

Your data center generates tens of terabytes of data daily and has a cumulative historic data volume of 5PB. The data center is running short of storage as well as bandwidth infrastructure to store or transfer this data. Later you would like to analyze this data using Redshift or Athena, however, first you must clean it using a proprietary process running on EC2.

What's the optimal way of moving this data to the cloud?

- Use S3 transfer acceleration
- Use AWS Data Migration
- Use Snowball Edge **(Correct)**
- Use Volume Gateway

Explanation

Correct option:

Use Snowball Edge

AWS Snowball, a part of the AWS Snow Family, is a data migration and edge computing device that comes in two options. Snowball Edge Storage Optimized devices provide both block storage and Amazon S3-compatible object storage, and 40 vCPUs. They are well suited for local storage and large scale data transfer. Snowball Edge Compute Optimized devices provide 52 vCPUs, block and object storage, and an optional GPU for use cases like advanced machine learning and full-motion video analysis in disconnected environments.

Snowball Edge Storage Optimized is the optimal choice if you need to securely and quickly transfer dozens of terabytes to petabytes of data to AWS. It provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity to address large scale data transfer and pre-processing use cases.

For the given use-case, you can use multiple Snowball Edge devices to migrate the entire data to AWS Cloud.

Exam Alert:

The original Snowball devices were transitioned out of service and Snowball Edge Storage Optimized are now the primary devices used for data transfer. You may see the Snowball device on the exam, just remember that the original Snowball device had 80TB of storage space.

Incorrect options:

Use S3 transfer acceleration - Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path. As the data center does not have sufficient bandwidth infrastructure, so this option is ruled out.

Use Volume Gateway - You can configure the AWS Storage Gateway service as a Volume Gateway to present cloud-based iSCSI block storage volumes to your on-premises applications. The Volume Gateway provides either a local cache or full volumes on-premises while also storing full copies of your volumes in the AWS cloud. Volume Gateway also provides Amazon EBS Snapshots of your data for backup, disaster recovery, and migration. It's easy to get started with the Volume Gateway: Deploy it as a virtual machine or hardware appliance, give it local disk resources, connect it to your applications, and start using your hybrid cloud storage for block data. As the data center does not have sufficient bandwidth infrastructure, so this option is ruled out.

How Storage Gateway Works:

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

Use AWS Data Migration - AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from the most widely used commercial and open-source databases. As the data center does not have sufficient bandwidth infrastructure, so this option is ruled out.

References:

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

https://aws.amazon.com/dms/

https://aws.amazon.com/storagegateway/

Question 7: **Correct**

Your website is hosted on S3 and exposed through a CloudFront distribution and some users are said to experience a lot of 501 errors.

How can you analyze these errors and come up with a solution?

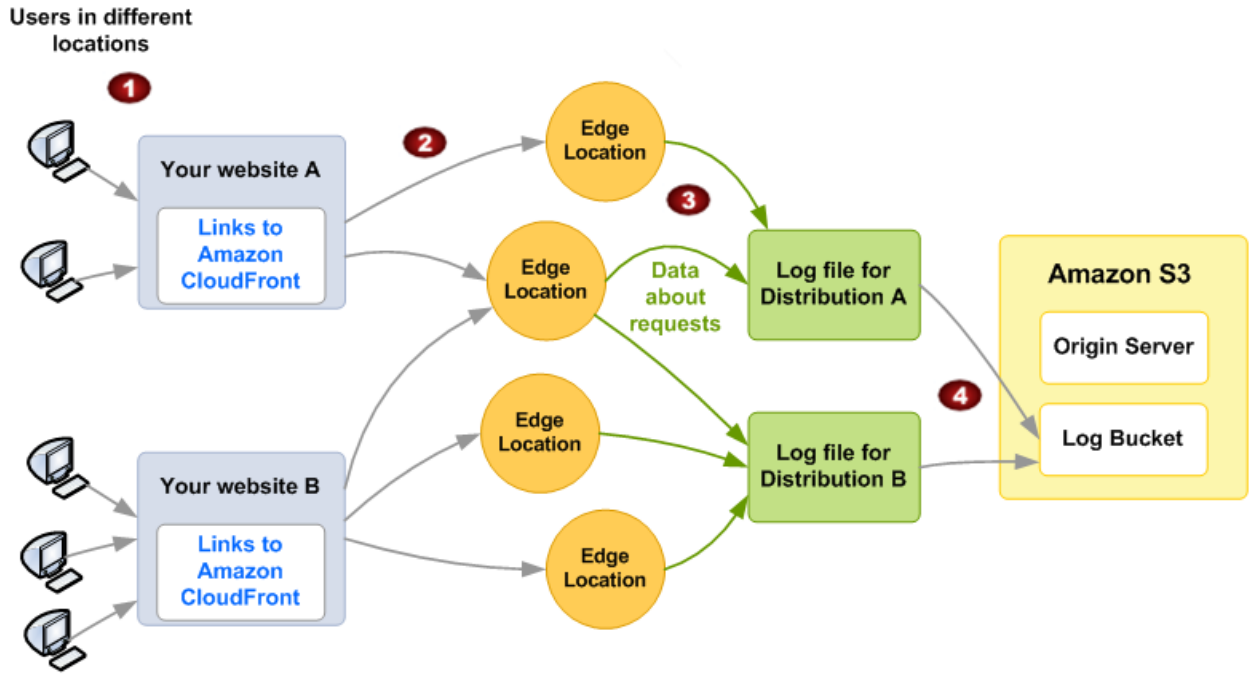
- ○ **Analyze the CloudFront access logs using Inspector**
- ○ **Enable S3 access logs and analyze using Athena**
- ○ **Enable S3 access logs and analyze using Inspector**
- ○ **Analyze the CloudFront access logs using Athena** **(Correct)**

Explanation

Correct option:

Analyze the CloudFront access logs using Athena

You can configure CloudFront to create log files that contain detailed information about every user request that CloudFront receives. These are called standard logs, also known as access logs. These standard logs are available for both web and RTMP distributions. If you enable standard logs, you can also specify the Amazon S3 bucket that you want CloudFront to save files in.



via - https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html

AWS recommends that you use the logs to understand the nature of the requests for your content, not as a complete accounting of all requests. CloudFront delivers access logs on a best-effort basis. The log entry for a particular request might be delivered long after the request was actually processed and, in rare cases, a log entry might not be delivered at all. When a log entry is omitted from access logs, the number of entries in the access logs won't match the usage that appears in the AWS usage and billing reports.

Incorrect options:

Analyze the CloudFront access logs using Inspector

Enable S3 access logs and analyze using Inspector

Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances.

Inspector cannot be used to analyze CloudFront access logs or S3 access logs, so both these options are incorrect.

Enable S3 access logs and analyze using Athena - The S3 access logs will not provide details about the user IP and other crucial information, as the requests are proxied through CloudFront. Additionally, results are cached in CloudFront and the S3 access logs won't contain a lot of information, so this option is incorrect.

Reference:

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

Question 8: **Correct**

You have an ASG in which the Terminate process is suspended. Your ASG goes into a rebalance, what will happen?

- ○ **The rebalance will start and the EC2 instances will fail to get launched**

- ○ **The rebalance will not start, as the terminate process is suspended**

- ○ **The rebalance will start and the EC2 instances will launch, the ASG will grow up to 10% of its size. After a bit, the instances will get terminated as the ASG is at overcapacity**

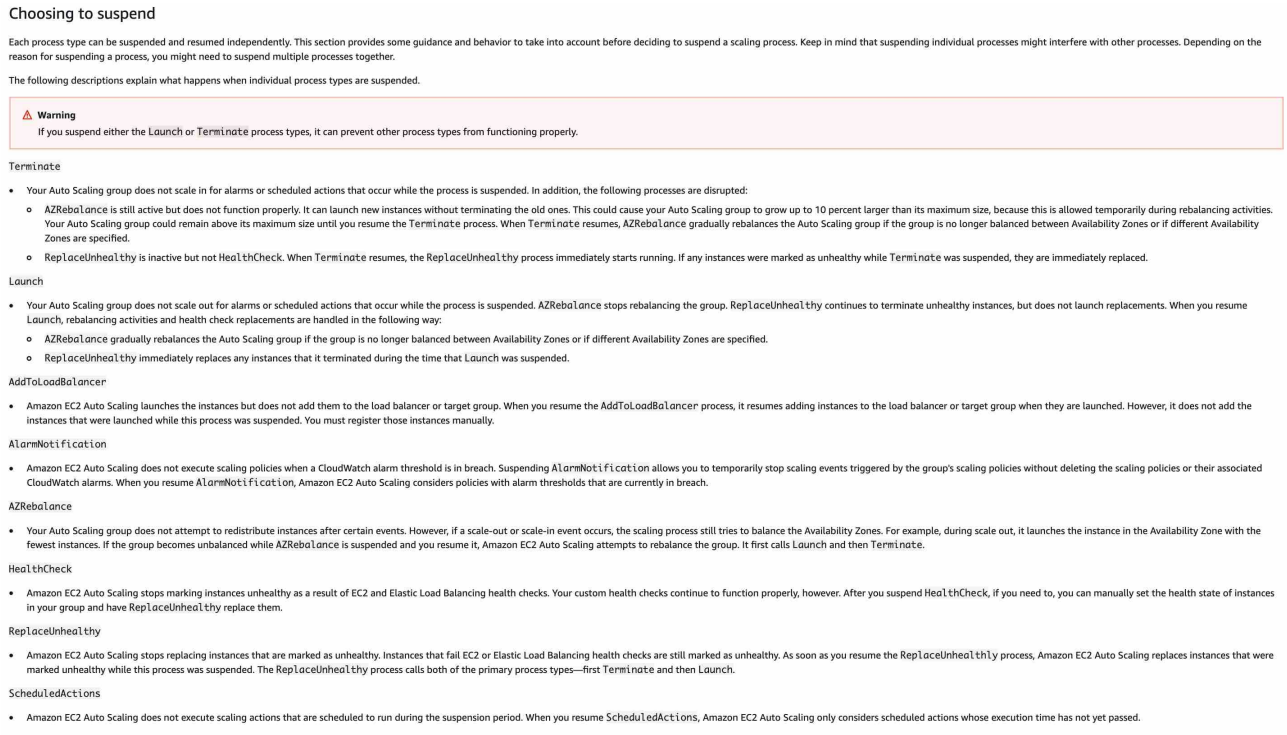
- ○ **The rebalance will start and the EC2 instances will launch, the ASG will grow up to 10% of its size. The instances will not get terminated** **(Correct)**

Explanation

Correct option:

The rebalance will start and the EC2 instances will launch, the ASG will grow up to 10% of its size. The instances will not get terminated

If the Terminate process is suspended, your Auto Scaling group does not scale in for alarms or scheduled actions that occur. While the Terminate process is suspended and the AZRebalance process is still active then AZRebalance will not function properly. AZRebalance will be able to launch new instances without terminating the old ones. This could cause your Auto Scaling group to grow up to 10 percent larger than its maximum size because this is allowed temporarily during rebalancing activities.



Choosing to suspend

Each process type can be suspended and resumed independently. This section provides some guidance and behavior to take into account before deciding to suspend a scaling process. Keep in mind that suspending individual processes might interfere with other processes. Depending on the reason for suspending a process, you might need to suspend multiple processes together.

The following descriptions explain what happens when individual process types are suspended.

⚠ **Warning**
If you suspend either the `Launch` or `Terminate` process types, it can prevent other process types from functioning properly.

`Terminate`

- Your Auto Scaling group does not scale in for alarms or scheduled actions that occur while the process is suspended. In addition, the following processes are disrupted:
  - `AZRebalance` is still active but does not function properly. It can launch new instances without terminating the old ones. This could cause your Auto Scaling group to grow up to 10 percent larger than its maximum size, because this is allowed temporarily during rebalancing activities. Your Auto Scaling group could remain above its maximum size until you resume the `Terminate` process. When `Terminate` resumes, `AZRebalance` gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
  - `ReplaceUnhealthy` is inactive but not `HealthCheck`. When `Terminate` resumes, the `ReplaceUnhealthy` process immediately starts running. If any instances were marked as unhealthy while `Terminate` was suspended, they are immediately replaced.

`Launch`

- Your Auto Scaling group does not scale out for alarms or scheduled actions that occur while the process is suspended. `AZRebalance` stops rebalancing the group. `ReplaceUnhealthy` continues to terminate unhealthy instances, but does not launch replacements. When you resume `Launch`, rebalancing activities and health check replacements are handled in the following way:
  - `AZRebalance` gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
  - `ReplaceUnhealthy` immediately replaces any instances that it terminated during the time that `Launch` was suspended.

`AddToLoadBalancer`

- Amazon EC2 Auto Scaling launches the instances but does not add them to the load balancer or target group. When you resume the `AddToLoadBalancer` process, it resumes adding instances to the load balancer or target group when they are launched. However, it does not add the instances that were launched while this process was suspended. You must register those instances manually.

`AlarmNotification`

- Amazon EC2 Auto Scaling does not execute scaling policies when a CloudWatch alarm threshold is in breach. Suspending `AlarmNotification` allows you to temporarily stop scaling events triggered by the group's scaling policies without deleting the scaling policies or their associated CloudWatch alarms. When you resume `AlarmNotification`, Amazon EC2 Auto Scaling considers policies with alarm thresholds that are currently in breach.

`AZRebalance`

- Your Auto Scaling group does not attempt to redistribute instances after certain events. However, if a scale-out or scale-in event occurs, the scaling process still tries to balance the Availability Zones. For example, during scale out, it launches the instance in the Availability Zone with the fewest instances. If the group becomes unbalanced while `AZRebalance` is suspended and you resume it, Amazon EC2 Auto Scaling attempts to rebalance the group. It first calls `Launch` and then `Terminate`.

`HealthCheck`

- Amazon EC2 Auto Scaling stops marking instances unhealthy as a result of EC2 and Elastic Load Balancing health checks. Your custom health checks continue to function properly, however. After you suspend `HealthCheck`, if you need to, you can manually set the health state of instances in your group and have `ReplaceUnhealthy` replace them.

`ReplaceUnhealthy`

- Amazon EC2 Auto Scaling stops replacing instances that are marked as unhealthy. Instances that fail EC2 or Elastic Load Balancing health checks are still marked as unhealthy. As soon as you resume the `ReplaceUnhealthly` process, Amazon EC2 Auto Scaling replaces instances that were marked unhealthy while this process was suspended. The `ReplaceUnhealthy` process calls both of the primary process types—first `Terminate` and then `Launch`.

`ScheduledActions`

- Amazon EC2 Auto Scaling does not execute scaling actions that are scheduled to run during the suspension period. When you resume `ScheduledActions`, Amazon EC2 Auto Scaling only considers scheduled actions whose execution time has not yet passed.

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html

Incorrect options:

The rebalance will not start, as the terminate process is suspended

The rebalance will start and the EC2 instances will fail to get launched

The rebalance will start and the EC2 instances will launch, the ASG will grow up to 10% of its size. After a bit, the instances will get terminated as the ASG is at overcapacity

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

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html

Question 9: **Correct**

Your company is experiencing an unusually high cost of Elastic IPs (EIPs) as most of them sit unassigned. Management would like to see a report showing the allocation of costs for these EIPs by department.

What do you advise on doing?

○ **Define an AWS Config Rule per department and track cost**

○ **Define Cost Allocation Tags and generate a report using Cost Explorer** **(Correct)**

○ **Use AWS Artifact to forbid people from leaving Elastic IPs unassigned for more than 20 minutes**

Create an AWS Lambda function that checks on an hourly basis the status of the EIPs and tracks using CloudTrail who is the last person who accessed them

Explanation

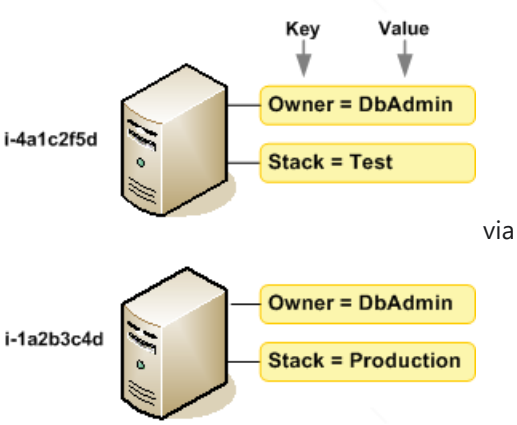
Correct option:

Define Cost Allocation Tags and generate a report using Cost Explorer

An Elastic IP address is a static, public, IPv4 address allocated to your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account. Elastic IPs do not change and remain allocated to your account until you delete them.

To ensure efficient use of Elastic IP addresses, AWS imposes a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance.

A tag is a label that you or AWS assigns to an AWS resource. Each tag consists of a key and a value. For each resource, each tag key must be unique, and each tag key can have only one value. You can use tags to organize your resources, and cost allocation tags to track your AWS costs on a detailed level. After you activate cost allocation tags, AWS uses the cost allocation tags to organize your resource costs on your cost allocation report, to make it easier for you to categorize and track your AWS costs.



- https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html

After you or AWS applies tags to your AWS resources (such as Amazon EC2 instances or Amazon S3 buckets) and you activate the tags in the Billing and Cost Management console, AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and costs grouped by your active tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

For the given use-case, you can define department-wise cost allocation tags for EC2 instances with EIPs and then generate a report using Cost Explorer.

Incorrect options:

Define an AWS Config Rule per department and track cost - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?". You cannot use AWS Config to track the cost of EC2 instances with EIPs.

Create an AWS Lambda function that checks on an hourly basis the status of the EIPs and tracks using CloudTrail who is the last person who accessed them - AWS Lambda lets you run code without provisioning or managing servers. This option has been added as a distractor, you cannot use AWS Config to track the cost of EC2 instances with EIPs.

Use AWS Artifact to forbid people from leaving Elastic IPs unassigned for more than 20 minutes - AWS Artifact is a self-service audit artifact retrieval portal that provides our customers with on-demand access to AWS' compliance documentation and AWS agreements. You cannot use AWS Artifact to track usage for unassigned Elastic IPs.

Reference:

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

https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html

Question 10: **Correct**

You run a full e-commerce website on Elastic Beanstalk, which provisions an Application Load Balancer in a public subnet, an Auto Scaling Group that spans 3 private subnets, and an RDS database in Multi-AZ mode in two private subnets. The Load Balancer can access your application, and your application can access the database.

Yet, you have trouble patching your EC2 instances using SSM as these instances cannot access the internet. What's the issue?

- ○ **Open up security groups on the EC2 instances**
- ○ **Deploy the instances in the public subnet instead. Private subnets cannot access the internet**
- ○ **Deploy an Internet Gateway in the public subnet and add entries to your route table**
- ○ **Deploy a NAT Gateway in the public subnet and add entries to your route table** **(Correct)**

Explanation

Correct option:

Deploy a NAT Gateway in the public subnet and add entries to your route table

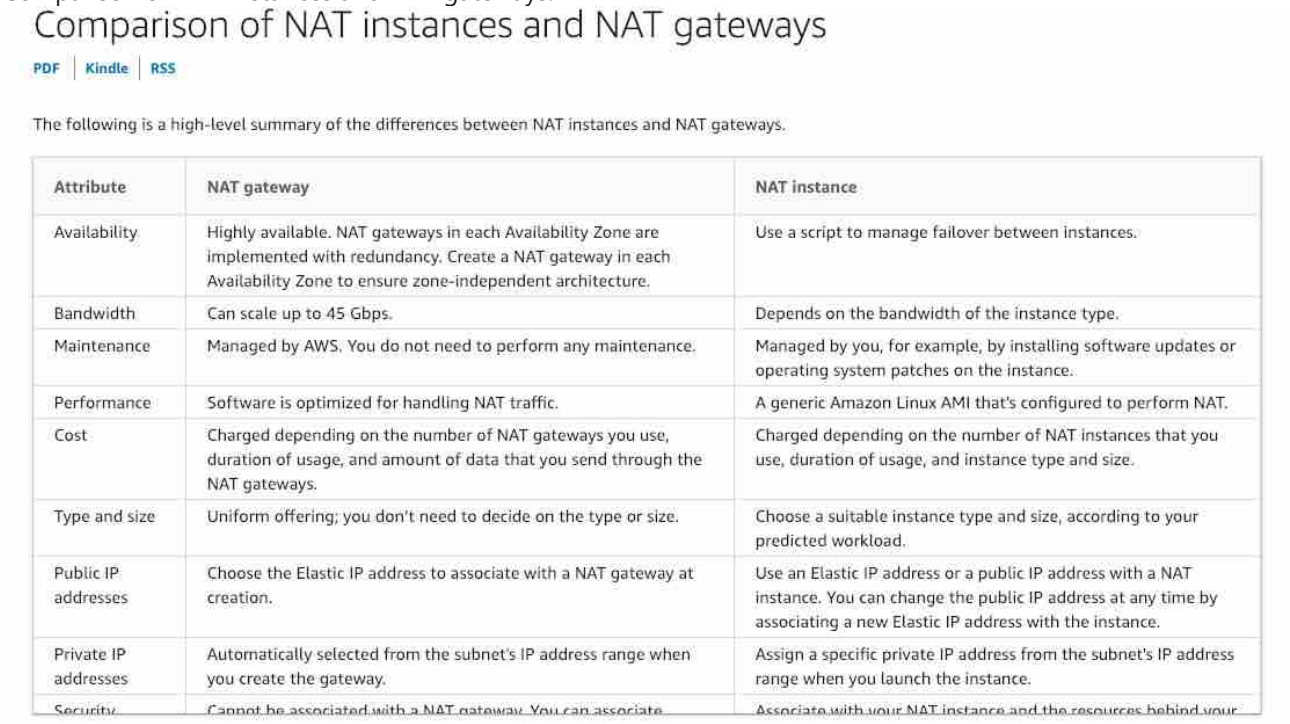
You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances. A NAT gateway has the following characteristics and limitations:

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

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

Therefore you must use a NAT Gateway in your public subnet in order to provide internet access to your instances in your private subnets. You also need to set up the appropriate entries in the route table of the private subnets. You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Comparison of NAT instances and NAT gateways:



Comparison of NAT instances and NAT gateways

PDF | Kindle | RSS

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

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

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

Incorrect options:

Deploy the instances in the public subnet instead. Private subnets cannot access the internet - Deploying the instances in private subnet would jeopardize the security of the EC2 instances, so this option is not correct.

Open up security groups on the EC2 instances - A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, we use the default security group. You can add rules to each security group that allows traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

Opening up the security groups would jeopardize the security of the EC2 instances, so this option is not correct.

Deploy an Internet Gateway in the public subnet and add entries to your route table - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. Internet Gateways must be deployed in a public subnet and the corresponding entry should be added to the route table.

This option has been added as a distractor as it just states the necessary conditions for a public subnet to have internet access.

Reference:

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

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

Question 11: **Correct**

Your company has decided to elect AWS champions that will train and drive the AWS cloud adoption internally. You would like to perform an analysis to see your most active AWS users.

How can you do that?

- ○ **Use IAM usage report and Athena**
- ○ **Use CloudTrail and Athena** **(Correct)**
- ○ **Use VPC Flow Logs and Athena**
- ○ **Use GuardDuty and Athena**

Explanation

Correct option:

Use CloudTrail and Athena

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

For an ongoing record of events in your AWS account, including events for IAM and AWS STS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify.

Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries.

You can use Athena to analyze the CloudTrail log data in S3 specific to IAM events.

Incorrect options:

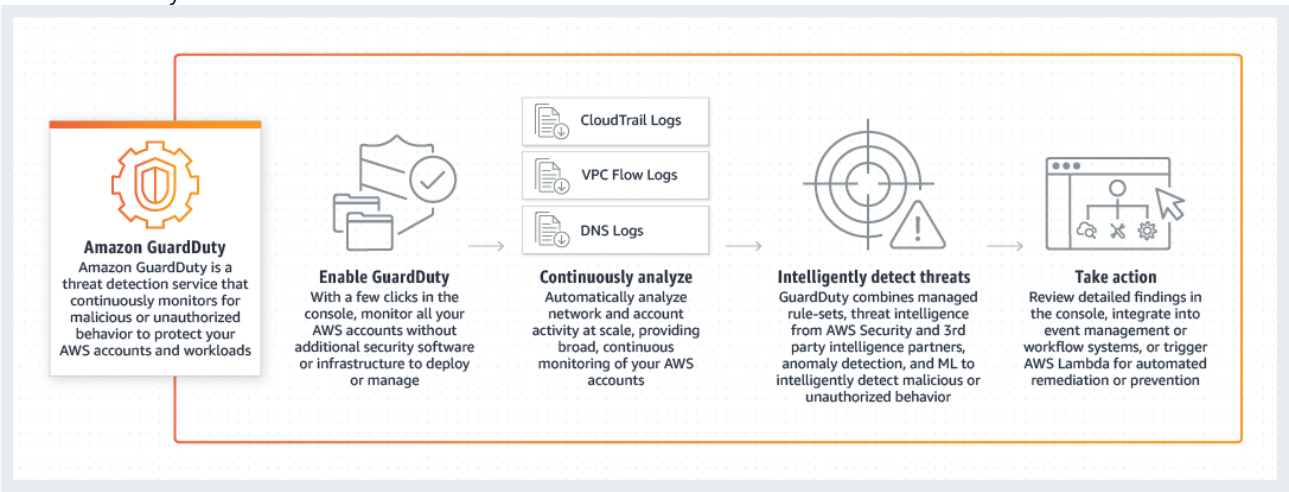
Use IAM usage report and Athena - This is a made-up option as there is no such thing as an IAM usage report.

Use VPC Flow Logs and Athena - VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data is used to analyze network traces and helps with network security. Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. You cannot use VPC Flow Logs to get information about the most active AWS users.

Use GuardDuty and Athena - GuardDuty is a threat detection service that monitors malicious activity and unauthorized behavior to protect your AWS account. GuardDuty analyzes billions of events across your AWS accounts from AWS CloudTrail (AWS user and API activity in your accounts), Amazon VPC Flow Logs (network

traffic data), and DNS Logs (name query patterns). GuardDuty cannot be used to get information about the most active AWS users.

How GuardDuty Works:



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

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/cloudtrail-integration.html

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

Question 12: **Correct**

As part of monitoring your global e-learning website, you have decided to implement a CloudWatch dashboard. The most important metric to monitor is the number of users that are connected over time, in each region.

Which option should you opt for?

- ○ **Create one CloudWatch dashboard and add a special widget of type multi-region graph**
- ○ **Create one CloudWatch dashboard per region**
- ○ **Create one CloudWatch dashboard of the metric, and tick the option "global metric". Use the CloudWatch Dashboard region dropdown to change the graph on demand**
- ○ **Create one CloudWatch dashboard and add a graph per region using the region selector in the top right corner of the AWS Console** **(Correct)**

Explanation

Correct option:

Create one CloudWatch dashboard and add a graph per region using the region selector in the top right corner of the AWS Console

Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources. To address the given use-case, you can create a graph per region using the region selector.

Incorrect options:

Create one CloudWatch dashboard per region - Since each CloudWatch dashboard supports resources from multiple regions, so this option is incorrect.

Create one CloudWatch dashboard and add a special widget of type: multi-region graph - There is no such thing as a special widget of type: multi-region graph. This option has been added as a distractor.

Create one CloudWatch dashboard of the metric, and tick the option "global metric". Use the CloudWatch Dashboard region dropdown to change the graph on demand - There is no such option as a "global metric". This option has been added as a distractor.

References:

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

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

Question 13: **Correct**

A company provides their on-premises applications with low latency access to data by maintaining all the data on-premises. With increased infrastructure costs and unreliable disaster recovery options for the on-premises infrastructure, the company wants to move the data to AWS Cloud while still maintaining the current low latency access for the on-premises applications.

Which is the right way to store the on-premises iSCSI block devices data on AWS Cloud?

- ○ **Use File Gateway of AWS Storage Gateway service**
- ○ **Use Amazon Elastic File System (Amazon EFS) to elastically scale for hundreds of compute instances at low latency**
- ○ **Use Volume Gateway of AWS Storage Gateway service** **(Correct)**
- ○ **Use Amazon Simple Storage Service (Amazon S3) web service interface to store and retrieve any amount of data, at any time**

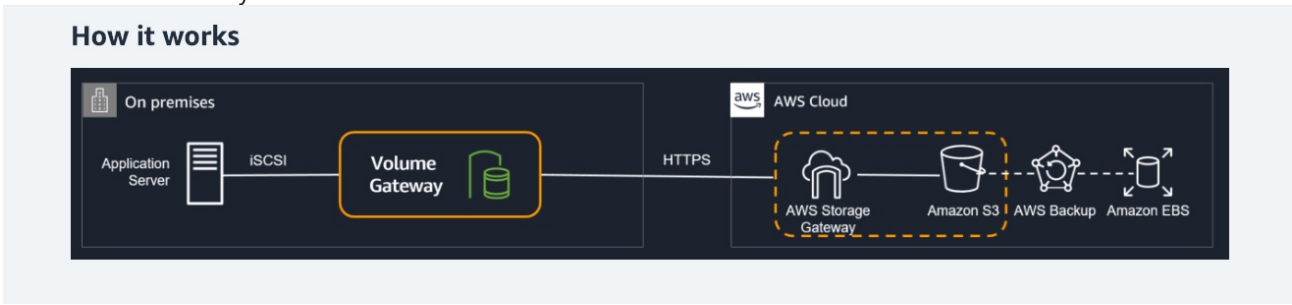
Explanation

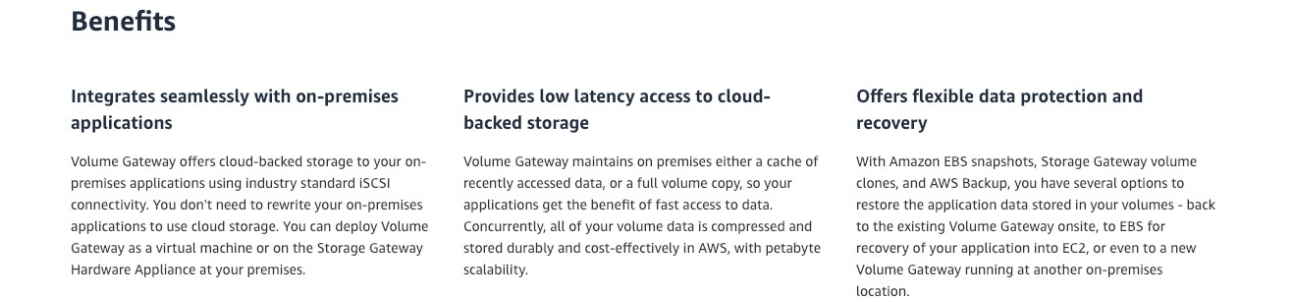
Correct option:

Use Volume Gateway of AWS Storage Gateway service - AWS Storage Gateway is a set of hybrid cloud services that gives you on-premises access to virtually unlimited cloud storage. Customers use Storage Gateway to integrate AWS Cloud storage with existing on-site workloads so they can simplify storage management and reduce costs for key hybrid cloud storage use cases.

Volume Gateway presents cloud-backed iSCSI block storage volumes to your on-premises applications. Volume Gateway stores and manages on-premises data in Amazon S3 on your behalf and operates in either cache mode or stored mode. In the cached Volume Gateway mode, your primary data is stored in Amazon S3, while retaining your frequently accessed data locally in the cache for low latency access. In the stored Volume Gateway mode, your primary data is stored locally and your entire dataset is available for low latency access on-premises while also asynchronously getting backed up to Amazon S3. In either mode, you can take point-in-time copies of your volumes using AWS Backup, which are stored in AWS as Amazon EBS snapshots. Using Amazon EBS Snapshots enables you to make space-efficient versioned copies of your volumes for data protection, recovery, migration, and various other copy data needs.

How Volume Gateway works:





via - https://aws.amazon.com/storagegateway/volume/

Incorrect options:

Use File Gateway of AWS Storage Gateway service - Amazon S3 File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. Amazon S3 File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises data-intensive Amazon EC2-based applications that need file protocol access to S3 object storage.

Use Amazon Elastic File System (Amazon EFS) to elastically scale for hundreds of compute instances at low latency - Amazon EFS provides shared access to data using a traditional file sharing permissions model and hierarchical directory structure via the NFSv4 protocol. Applications that access data using a standard file system interface provided through the operating system can use Amazon EFS to take advantage of the scalability and reliability of file storage in the cloud without writing any new code or adjusting applications.

Use Amazon Simple Storage Service (Amazon S3) web service interface to store and retrieve any amount of data, at any time - Amazon S3 is an object storage platform that uses a simple API for storing and accessing data. Applications that do not require a file system structure and are designed to work with object storage can use Amazon S3 as a massively scalable, durable, low-cost object storage solution.

Reference:

https://aws.amazon.com/storagegateway/volume/

Question 14: **Correct**

You distribute a monthly raw data extract of your public forum's discussions that is about 10TB each month. Currently, the archive is distributed through an EFS drive, that is mounted on all your EC2 instances. Customers retrieve the file through the load balancer you have. This solution is costing you a lot of money and forces you to tremendously scale on the 1st of each month as people all try to retrieve the file at the same time.

What can you do to improve the situation?

- ○ **Store the files on instance stores instead, so you don't need to use EFS anymore**
- ○ **Enable enhanced networking between EC2 and ALB**
- ○ **Store the files in S3 and distribute them using a CloudFront distribution instead** **(Correct)**
- ○ **Enable static file caching on the ALB**

Explanation

Correct option:

Store the files in S3 and distribute them using a CloudFront distribution instead

S3 is more cost-effective than EFS. For example, per GB storage cost for S3 is $0.023/month whereas per GB storage cost for EFS is $0.3/month. Further, storing your static content with S3 provides a lot of advantages. But to help optimize your application's performance and security while effectively managing cost, AWS recommends that you also set up Amazon CloudFront to work with your S3 bucket to serve and protect the content. CloudFront is a content delivery network (CDN) service that delivers static and dynamic web content, video streams, and APIs around the world, securely and at scale. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.

Incorrect options:

Enable static file caching on the ALB - This is a distractor as there is no such thing as static file caching on the ALB.

Store the files on instance stores instead, so you don't need to use EFS anymore - You cannot use Instance Stores since they are physically attached to their own EC2 instances. Instance Store is not a shared storage like EFS, so this option is ruled out.

Enable enhanced networking between EC2 and ALB - Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported EC2 instance types. SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces. There is no such thing as enhanced networking between EC2 and ALB.

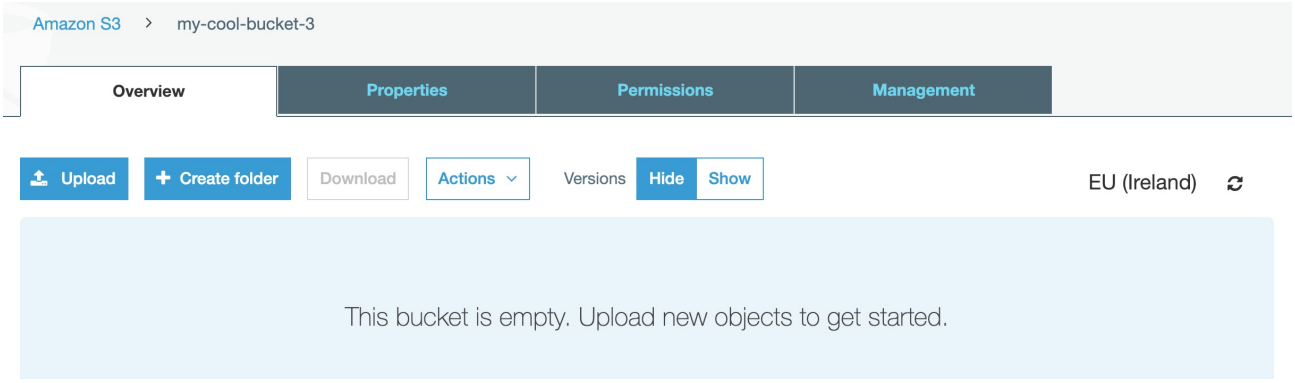
References:

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html

Question 15: **Correct**

You are in S3 and have deleted all the files in it. As you can see, the bucket is empty:



You have tried to delete the bucket afterward and it fails with an error saying the bucket is not empty.

What's the issue?

- ◯ **S3 versioning is enabled and delete markers are still present in the bucket** **(Correct)**
- ◯ **S3 is eventually consistent. Wait two minutes and retry, it will work then**
- ◯ **Some files are in Glacier**
- ◯ **An S3 bucket policy is set up and it prevents bucket deletion**

Explanation

Correct option:

S3 versioning is enabled and delete markers are still present in the bucket

S3 Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. When you enable versioning for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.

If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

If you delete an object, instead of removing it permanently, Amazon S3 inserts a delete marker, which becomes the current object version.

So for the given use-case, as delete markers are still present in the bucket, therefore you get the error saying the bucket is not empty.

Incorrect options:

S3 is eventually consistent. Wait two minutes and retry, it will work then - This is a made-up option. Amazon S3 provides strong read-after-write consistency for PUTs and DELETEs of objects in your Amazon S3 bucket in all AWS Regions.

An S3 bucket policy is set up and it prevents bucket deletion - You could set up a bucket policy to prevent bucket deletion, but it would not present an error that says the bucket is not empty.

Some files are in Glacier - You store your data in Amazon S3 Glacier as archives. Archives may be further grouped into vaults. So files are not stored in buckets for Glacier. So while deleting, you will not get an error that says the bucket is not empty.

Reference:

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

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

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

Question 16: **Correct**

When your baby products website started, it was running at low volume so your instances of type T2.micro were doing a fine job. After a while, your website exploded in popularity and now your ELB is seeing greater traffic. You had planned for the scaling events and your T2.micro instances are running in an auto scaling group. You also noticed that the EC2 instances are experiencing high CPU utilization because the CPU is being throttled and has very poor performance and your users are complaining. Hence the ASG is not scaling.

What can you do to improve the performance of your application? (Select two)

Your Load Balancer needs to be pre-warmed, and then your users will be happy

Change the ELB from an Application Load Balancer type to a Network Load Balancer type

Your T2.micro instances have run out of burst credit. Switch to a T2.large or m4.large instance type for greater stability (Correct)

You need to disable ELB stickiness

You should enable T2 unlimited (Correct)

Explanation

Correct options:

Traditional Amazon EC2 instance types provide fixed CPU utilization, while burstable performance instances provide a baseline level of CPU utilization with the ability to burst CPU utilization above the baseline level. The baseline utilization and ability to burst are governed by CPU credits.

The CPU credits used depends on CPU utilization. The following scenarios all use one CPU credit:

One vCPU at 100% utilization for one minute

One vCPU at 50% utilization for two minutes

Two vCPUs at 25% utilization for two minutes

Burstable performance instances are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. Burstable performance instances are well suited for a wide range of general-purpose applications. Examples include microservices, low-latency interactive applications, small and medium databases, virtual desktops, development, build, and stage environments, code repositories, and product prototypes.

The given use-case highlights the fact that the instances are being throttled for CPU usage. To resolve CPU throttling, you can either enable T2/T3 Unlimited, or change the instance type.

Your T2.micro instances have run out of burst credit. Switch to a T2.large or m4.large instance type for greater stability

You can change to one of the following instance types:

A T2 or T3 instance type with a higher CPU credit limit.

An instance type that doesn't use a CPU credit bucket model.

You should enable T2 unlimited

T2 Unlimited instances can sustain high CPU performance for as long as a workload needs it. For most general-purpose workloads, T2 Unlimited instances will provide ample performance without any additional charges. If the instance needs to run at higher CPU utilization for a prolonged period, it can also do so at a flat additional rate of 5 cents per vCPU-hour.

Incorrect options:

Your Load Balancer needs to be pre-warmed, and then your users will be happy - ELB can handle the vast majority of use cases for the customers without requiring "pre-warming" (configuring the load balancer to have the appropriate level of capacity based on expected traffic). In certain scenarios, such as when flash traffic is expected, or in the case where a load test cannot be configured to gradually increase traffic, AWS recommends that you contact AWS to have your load balancer "pre-warmed". AWS will then configure the load balancer to have the appropriate level of capacity based on the traffic that you expect. AWS will need to know the start and end dates of your tests or expected flash traffic, the expected request rate per second and the total size of the typical request/response that you will be testing.

You need to disable ELB stickiness - You can use the sticky session feature (also known as session affinity), which enables the load balancer to bind a user's session to a specific instance.

Change the ELB from an Application Load Balancer type to a Network Load Balancer type

These three options have been added as distractors since the constraint is with CPU throttling for the EC2 instances and not at the ELB level.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/ec2-cpu-utilization-throttled/

https://aws.amazon.com/ec2/instance-types/t2/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-credits-baseline-concepts.html

https://aws.amazon.com/articles/best-practices-in-evaluating-elastic-load-balancing/#pre-warming

Question 17: **Correct**

You suspect some of your employees try to access files in S3 that they don't have access to.

How can you verify this is indeed the case without them noticing?

- ○ **Use a bucket policy**
- ○ **Restrict their IAM policies and look at CloudTrail logs**
- ○ **Enable S3 Access Logs and analyze them using Athena** **(Correct)**
- ○ **Use AWS Config to define compliance rules on these users**

Explanation

Correct option:

Enable S3 Access Logs and analyze them using Athena

By default, Amazon Simple Storage Service (Amazon S3) doesn't collect server access logs. When you enable logging, Amazon S3 delivers access logs for a source bucket to a target bucket that you choose. The target bucket must be in the same AWS Region as the source bucket and must not have a default retention period configuration.

Server access logging provides detailed records for the requests that are made to an S3 bucket. Server access logs are useful for many applications. For example, access log information can be useful in security and access audits. It can also help you learn about your customer base and understand your Amazon S3 bill. Each access log record provides details about a single access request, such as the requester, bucket name, request time, request action, response status, and an error code, if relevant.

Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries.

For the given use-case, you can enable S3 access logs and then use Athena to analyze the access patterns for specific employees.

Incorrect options:

Restrict their IAM policies and look at CloudTrail logs - Restricting their IAM policies would deny access to S3 which is to be avoided per the use-case.

Use a bucket policy - You cannot use a bucket policy to log S3 access information

Use AWS Config to define compliance rules on these users - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?". You cannot use AWS Config to log S3 access information.

Reference:

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

Question 18: **Correct**

The Big Data team at an insurance company is performing a nightly ETL on top of your production RDS database to compute a view and then extract it into their data lake in Amazon S3. This query has been performing reasonably well in your website's infancy but now that it has grown in popularity, the query is running for a much longer period and affects the user experience while they browse your website.

How can you improve the situation in the short and long term?

- ◯ **Upgrade the RDS instance type**
- ◯ **Use Athena to query RDS**
- ◯ **Enable RDS Multi-AZ**

◯ **Create an RDS Read Replica for the ETL team** **(Correct)**

Explanation

Correct option:

Create an RDS Read Replica for the ETL team

Amazon RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances. They make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. For the MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance. It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance.

For the given use-case, you can use one or more read replicas for the given source DB instance as the source for the ETL process to populate the data lake on S3.

Overview of Amazon RDS read replicas

Deploying one or more read replicas for a given source DB instance might make sense in a variety of scenarios, including the following:

- Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads. You can direct this excess read traffic to one or more read replicas.
- Serving read traffic while the source DB instance is unavailable. In some cases, your source DB instance might not be able to take I/O requests, for example due to I/O suspension for backups or scheduled maintenance. In these cases, you can direct read traffic to your read replicas. For this use case, keep in mind that the data on the read replica might be "stale" because the source DB instance is unavailable.
- Business reporting or data warehousing scenarios where you might want business reporting queries to run against a read replica, rather than your production DB instance.
- Implementing disaster recovery. You can promote a read replica to a standalone instance as a disaster recovery solution if the primary DB instance fails.

By default, a read replica is created with the same storage type as the source DB instance. However, you can create a read replica that has a different storage type from the source DB instance based on the options listed in the following table.

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

Incorrect options:

Enable RDS Multi-AZ - Amazon RDS Multi-AZ deployments provide enhanced availability and durability for RDS database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). You cannot use Multi-AZ to improve the ETL process as it cannot use the standby instance as a source for the ETL process.

Exam Alert:

Please review the key differences between Read Replicas and Multi-AZ:

Read replicas, Multi-AZ deployments, and multi-region deployments

Amazon RDS read replicas complement Multi-AZ deployments. While both features maintain a second copy of your data, there are differences between the two:

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

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

Upgrade the RDS instance type - Upgrade the RDS instance type may help a little bit, but the problem will resurface as traffic increases further. A better solution is to use the Read Replica as the source for the ETL process to populate the data lake on S3.

Use Athena to query RDS - Although Athena can query data from RDS by using its federated query feature, however, the problem would persist as the entire ETL load will fall on the main database. A better solution is to use the Read Replica as the source for the ETL process to populate the data lake on S3.

References:

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

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

https://aws.amazon.com/blogs/big-data/query-any-data-source-with-amazon-athenas-new-federated-query/

Question 19: **Correct**

As part of the best practices for DevOps, all your infrastructure is deployed using CloudFormation. This includes EBS volumes. When the CloudFormation stacks are deleted, it is mandatory to keep a snapshot of the EBS volumes for backup and compliance purposes.

How can you achieve this using CloudFormation?

◯ **Use DeletionPolicy=Snapshot** **(Correct)**

◯ **Enable termination protection**

○ **Use cfn helper scripts and Wait Conditions upon stack deletion**

○ **Reference the EBS volume as a stack output**

Explanation

Correct option:

Use DeletionPolicy=Snapshot

To control how AWS CloudFormation handles the EBS volume when the stack is deleted, set a deletion policy for your volume. You can choose to retain the volume, to delete the volume, or to create a snapshot of the volume.

Here is the sample YAML:

```
NewVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 100
    Encrypted: true
    AvailabilityZone: !GetAtt Ec2Instance.AvailabilityZone
    Tags:
      - Key: MyTag
        Value: TagValue
  DeletionPolicy: Snapshot
```

AWS::EC2::Volume

PDF | Kindle | RSS

Filter View All

Specifies an Amazon Elastic Block Store (Amazon EBS) volume.

When you use AWS CloudFormation to update an Amazon EBS volume that modifies `Iops`, `Size`, or `VolumeType`, there is a cooldown period before another operation can occur. This can cause your stack to report being in `UPDATE_IN_PROGRESS` or `UPDATE_ROLLBACK_IN_PROGRESS` for long periods of time.

Amazon EBS does not support sizing down an Amazon EBS volume. AWS CloudFormation will not attempt to modify an Amazon EBS volume to a smaller size on rollback.

Some common scenarios when you might encounter a cooldown period for Amazon EBS include:

- You successfully update an Amazon EBS volume and the update succeeds. When you attempt another update within the cooldown window, that update will be subject to a cooldown period.
- You successfully update an Amazon EBS volume and the update succeeds but another change in your `update-stack` call fails. The rollback will be subject to a cooldown period.

For more information on the cooldown period, see Requirements for Modifying EBS Volumes.

To control how AWS CloudFormation handles the volume when the stack is deleted, set a deletion policy for your volume. You can choose to retain the volume, to delete the volume, or to create a snapshot of the volume. For more information, see DeletionPolicy Attribute.

Note

If you set a deletion policy that creates a snapshot, all tags on the volume are included in the snapshot.

via - https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-ebs-volume.html

Incorrect options:

Enable termination protection - You can prevent a stack from being accidentally deleted by enabling termination protection on the stack. If a user attempts to delete a stack with termination protection enabled, the deletion fails and the stack--including its status--remains unchanged. You can enable termination protection on a stack when you create it. Termination protection on stacks is disabled by default. You can set termination protection on a stack with any status except DELETE_IN_PROGRESS or DELETE_COMPLETE.

Use cfn helper scripts and Wait Conditions upon stack deletion - The cfn helper scripts such as cfn-init, cfn-signal, etc help in installing packages or to indicate whether Amazon EC2 instances have been successfully created or updated. You cannot use these scripts to mandatorily keep a snapshot of the EBS volume.

Reference the EBS volume as a stack output - The optional Outputs section for a CloudFormation stack declares output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console. You should note that a stack that is referenced by another stack cannot be deleted and it cannot modify or remove the exported value. Just by referencing the EBS volume as a stack output, you will not be able to enforce the snapshot of the EBS volume.

References:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-ebs-volume.html

https://aws.amazon.com/blogs/aws/aws-cloudformation-update-yaml-cross-stack-references-simplified-substitution/

Question 20: **Correct**

A development team working for a gaming company has deployed an application on EC2 and needs CloudWatch monitoring for the relevant metrics with a resolution of 1 minute in order to set alarms that can rapidly react to changes.

As a SysOps Administrator, which of the following would you suggest as the MOST optimal solution?

- Use Systems Manager
- Enable EC2 detailed monitoring (Correct)
- Use AWS Lambda to retrieve metrics often using the application `/health` route
- The development team should create and send a high-resolution custom metric

Explanation

Correct option:

Enable EC2 detailed monitoring

Metrics are the fundamental concept in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch. Think of a metric as a variable to monitor, and the data points as representing the values of that variable over time.

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. So you can use EC2 detailed monitoring for the given use-case.

Enable or turn off detailed monitoring for your instances

PDF | Kindle | RSS

By default, your instance is enabled for basic monitoring. You can optionally enable detailed monitoring. After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.

The following describes the data interval and charge for basic and detailed monitoring for instances.

Basic monitoring

Data is available automatically in 5-minute periods at no charge.

Detailed monitoring

Data is available in 1-minute periods for an additional charge.

To get this level of data, you must specifically enable it for the instance. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances.

Charges for detailed monitoring

If you enable detailed monitoring, you are charged per metric that is sent to CloudWatch. You are not charged for data storage. For more information about pricing for detailed monitoring, see **Paid tier** on the Amazon CloudWatch pricing page. For a pricing example, see **Example 1 - EC2 Detailed Monitoring** on the Amazon CloudWatch pricing page.

via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-new.html

Incorrect options:

The development team should create and send a high-resolution custom metric - You can publish your own metrics to CloudWatch using the AWS CLI or an API. You can view statistical graphs of your published metrics with the AWS Management Console. Metrics produced by AWS services are standard resolution by default. When you publish a custom metric, you can define it as either standard resolution or high resolution. When you publish a high-resolution metric, CloudWatch stores it with a resolution of 1 second, and you can read and retrieve it with a period of 1 second, 5 seconds, 10 seconds, 30 seconds, or any multiple of 60 seconds. Custom metrics need extra effort to capture and push the custom metrics to CloudWatch via the API or CLI, so it's not the MOST optimal solution for the given use-case.

Use AWS Lambda to retrieve metrics often using the application `/health` route - This option has been added as a distractor as you cannot retrieve performance metrics using the `/health` route via Lambda or otherwise.

Use Systems Manager - Using AWS Systems Manager, you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources. You cannot use the Systems Manager to capture metrics for monitoring on CloudWatch.

References:

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-new.html

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

Question 21: **Correct**

A company has 90% of their server instances on AWS Cloud and the rest are provisioned in an on-premises data center. The company wants a tool/service that can collect metadata of these instances to validate the software running on the instances along with the configurations against their software policy.

Which of the following is the right fit for this requirement?

- ○ **Unified CloudWatch Agent**
- ○ **AWS Batch**
- ○ **AWS Systems Manager Inventory** **(Correct)**
- ○ **AWS Systems Manager Patch Manager**

Explanation

Correct option:

AWS Systems Manager Inventory

AWS Systems Manager Inventory provides visibility into your Amazon EC2 and on-premises computing environment. You can use Inventory to collect metadata from your managed instances. You can store this metadata in a central Amazon Simple Storage Service (Amazon S3) bucket, and then use built-in tools to query the data and quickly determine which instances are running the software and configurations required by your software policy, and which instances need to be updated. You can configure Inventory on all of your managed instances by using a one-click procedure. You can also configure and view inventory data from multiple AWS Regions and AWS accounts.

If the pre-configured metadata types collected by Systems Manager Inventory don't meet your needs, then you can create custom inventory. Custom inventory is simply a JSON file with information that you provide and add to the managed instance in a specific directory. When Systems Manager Inventory collects data, it captures this custom inventory data.

Systems Manager Inventory collects only metadata from your managed instances. Inventory doesn't access proprietary information or data.

Incorrect options:

AWS Systems Manager Patch Manager - Patch Manager, a capability of AWS Systems Manager, automates the process of patching managed instances with both security-related and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications. (On Windows Server, application support is limited to updates for applications released by Microsoft.) You can use Patch Manager to install Service Packs on Windows instances and perform minor version upgrades on Linux instances. You can patch fleets of Amazon Elastic Compute Cloud (Amazon EC2) instances or your on-premises servers and virtual machines (VMs) by operating system type.

AWS Config - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines.

You can use Inventory with AWS Config to audit your application configurations over time.

Unified CloudWatch Agent - The unified CloudWatch Agent enables you to collect internal system-level metrics from Amazon EC2 instances across operating systems, Collect system-level metrics from on-premises servers, retrieve custom metrics from your applications or services and collect logs from Amazon EC2 instances and on-premises servers.

Reference:

https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-inventory.html

Question 22: **Correct**

Amazon DynamoDB is experiencing a high level of rejected requests and this outage is directly impacting your applications. Your CTO would like to know all the resources that are affected in your AWS Account and how to mitigate them.

Where should you look first?

- ◯ **In AWS Personal Health Dashboard** **(Correct)**
- ◯ **In AWS Service Health Dashboard**
- ◯ **In DynamoDB**
- ◯ **In AWS Organizations**

Explanation

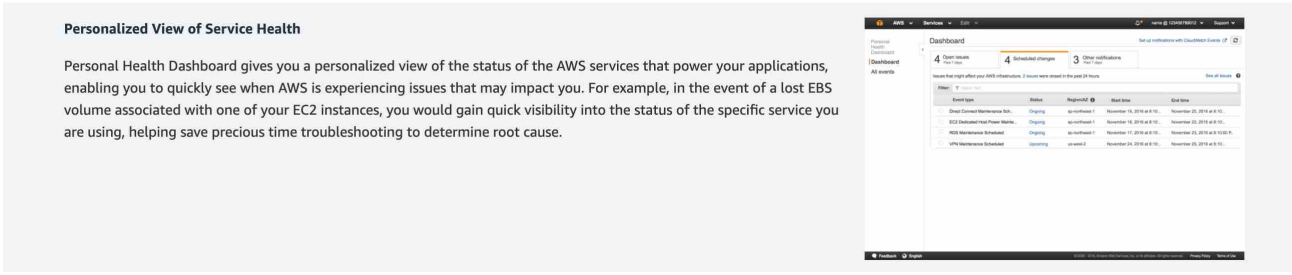
Correct option:

In AWS Personal Health Dashboard

AWS Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact you. While the Service Health Dashboard displays the general status of AWS services, Personal Health Dashboard gives you a personalized view into the performance and availability of the AWS services underlying your AWS resources.

What's more, Personal Health Dashboard proactively notifies you when AWS experiences any events that may affect you, helping provide quick visibility and guidance to help you minimize the impact of events in progress, and plan for any scheduled changes, such as AWS hardware maintenance.

AWS Personal Health Dashboard Overview:

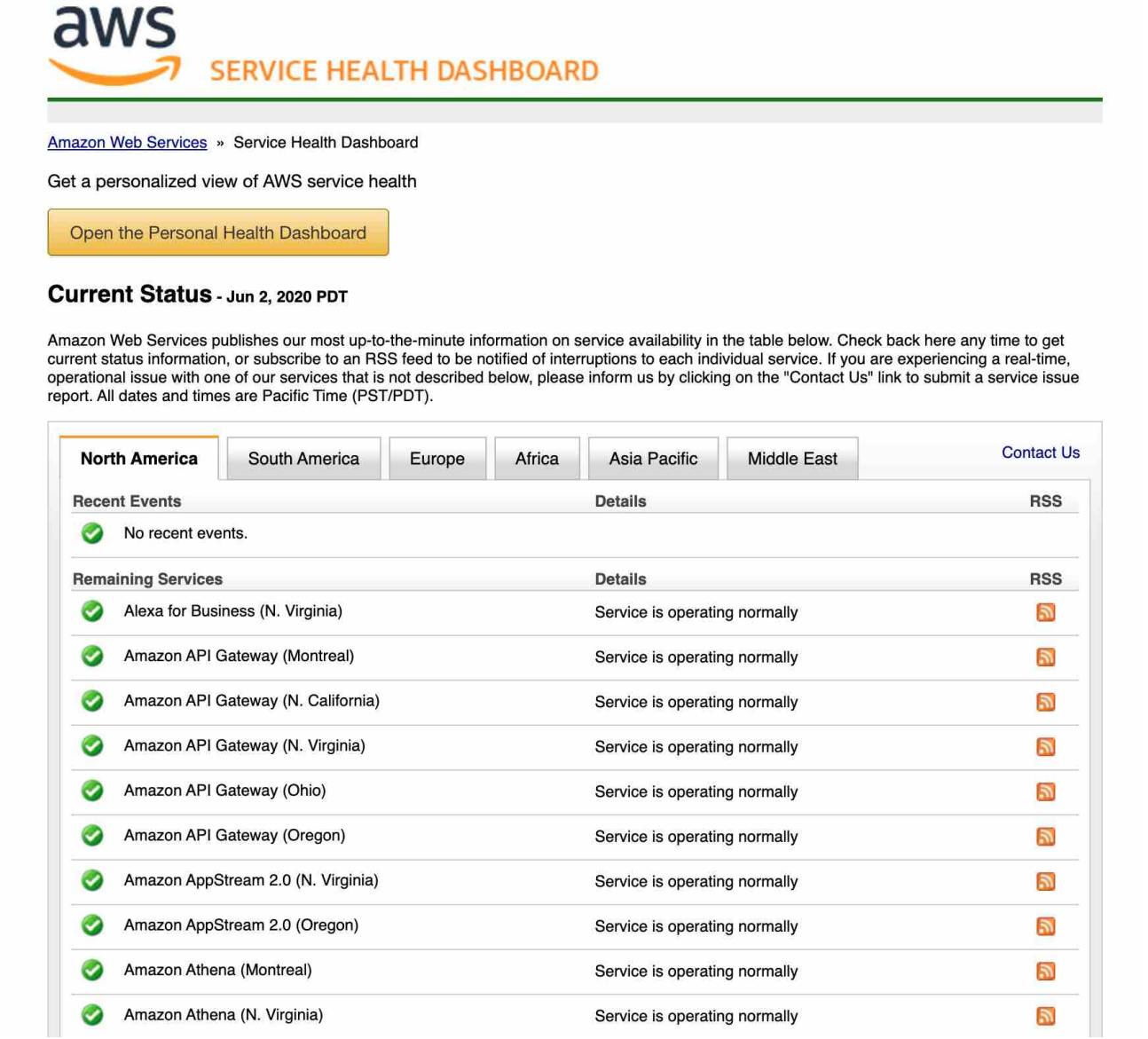


Incorrect options:

In DynamoDB - You cannot use DynamoDB to know about issues impacting all the resources in your AWS account.

In AWS Service Health Dashboard - AWS Service Health Dashboard publishes the most up-to-the-minute information on the status and availability of all AWS services in tabular form for all Regions that AWS is present in. You can check on this page https://status.aws.amazon.com/ to get current status information.

AWS Service Health Dashboard Overview:



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

In AWS Organizations - AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS. Organizations help you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts. You cannot use AWS Organizations to know about issues impacting all the resources in your AWS account.

References:

https://aws.amazon.com/premiumsupport/technology/personal-health-dashboard/

https://status.aws.amazon.com/

Question 23: **Correct**

You would like to establish a software only private connection between your corporate data center and your AWS VPC.

Which of the following component should you NOT use?

◯ **Direct Connect** **(Correct)**

Customer Gateway

Site-to-Site VPN

Virtual Private Gateway

Explanation

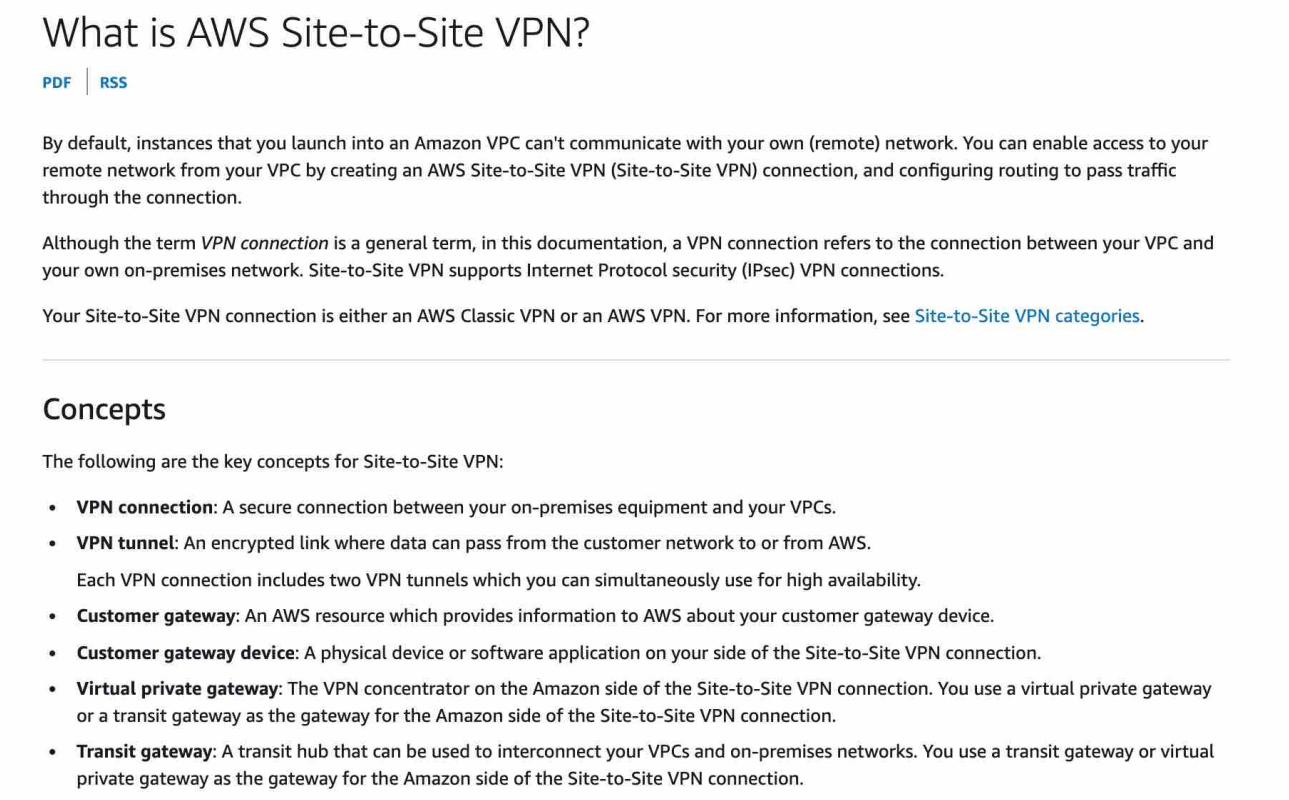
Correct option:

"Direct Connect"

AWS Direct Connect creates a dedicated private connection from a remote network to your VPC. This is a private connection and does not use the public internet. Takes at least a month to establish this connection. Direct Connect is a connectivity service and you cannot use it to provide AWS Cloud based storage access to on-premises applications. Direct Connect is a physical connection, hence it should NOT be used for the given use-case.

Incorrect options:

"Site-to-Site VPN" - By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection, and configuring routing to pass traffic through the connection.



What is AWS Site-to-Site VPN?

PDF | RSS

By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection, and configuring routing to pass traffic through the connection.

Although the term *VPN connection* is a general term, in this documentation, a VPN connection refers to the connection between your VPC and your own on-premises network. Site-to-Site VPN supports Internet Protocol security (IPsec) VPN connections.

Your Site-to-Site VPN connection is either an AWS Classic VPN or an AWS VPN. For more information, see Site-to-Site VPN categories.

Concepts

The following are the key concepts for Site-to-Site VPN:

- **VPN connection**: A secure connection between your on-premises equipment and your VPCs.
- **VPN tunnel**: An encrypted link where data can pass from the customer network to or from AWS.

  Each VPN connection includes two VPN tunnels which you can simultaneously use for high availability.
- **Customer gateway**: An AWS resource which provides information to AWS about your customer gateway device.
- **Customer gateway device**: A physical device or software application on your side of the Site-to-Site VPN connection.
- **Virtual private gateway**: The VPN concentrator on the Amazon side of the Site-to-Site VPN connection. You use a virtual private gateway or a transit gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.
- **Transit gateway**: A transit hub that can be used to interconnect your VPCs and on-premises networks. You use a transit gateway or virtual private gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.

via - https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html

"Virtual Private Gateway" - A Virtual Private Gateway is the Amazon VPC side of a VPN connection.

"Customer Gateway" - An AWS resource that provides information to AWS about your customer gateway device.

Reference:

https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html

Question 24: **Correct**

You have designed an AMI in an account that is optimizing the legacy database technology your gambling company has developed. You wish to share that AMI with other AWS accounts that belong to the same organization.

How do you do it?

- ○ **Edit the account list that can see the AMI from the AMI Console UI, and create an IAM role to be assumed by the other account using STS and the other accounts can start using it**
- ○ **The AMI can be shared without doing anything special. Just provide the target account with your secret AMI id and they can start using it**
- ○ **Create an IAM role to be assumed by the other account using STS and they can start accessing your AMI**
- ○ **Edit the account list that can see the AMI from the AMI Console UI and the other accounts can start using it** **(Correct)**

Explanation

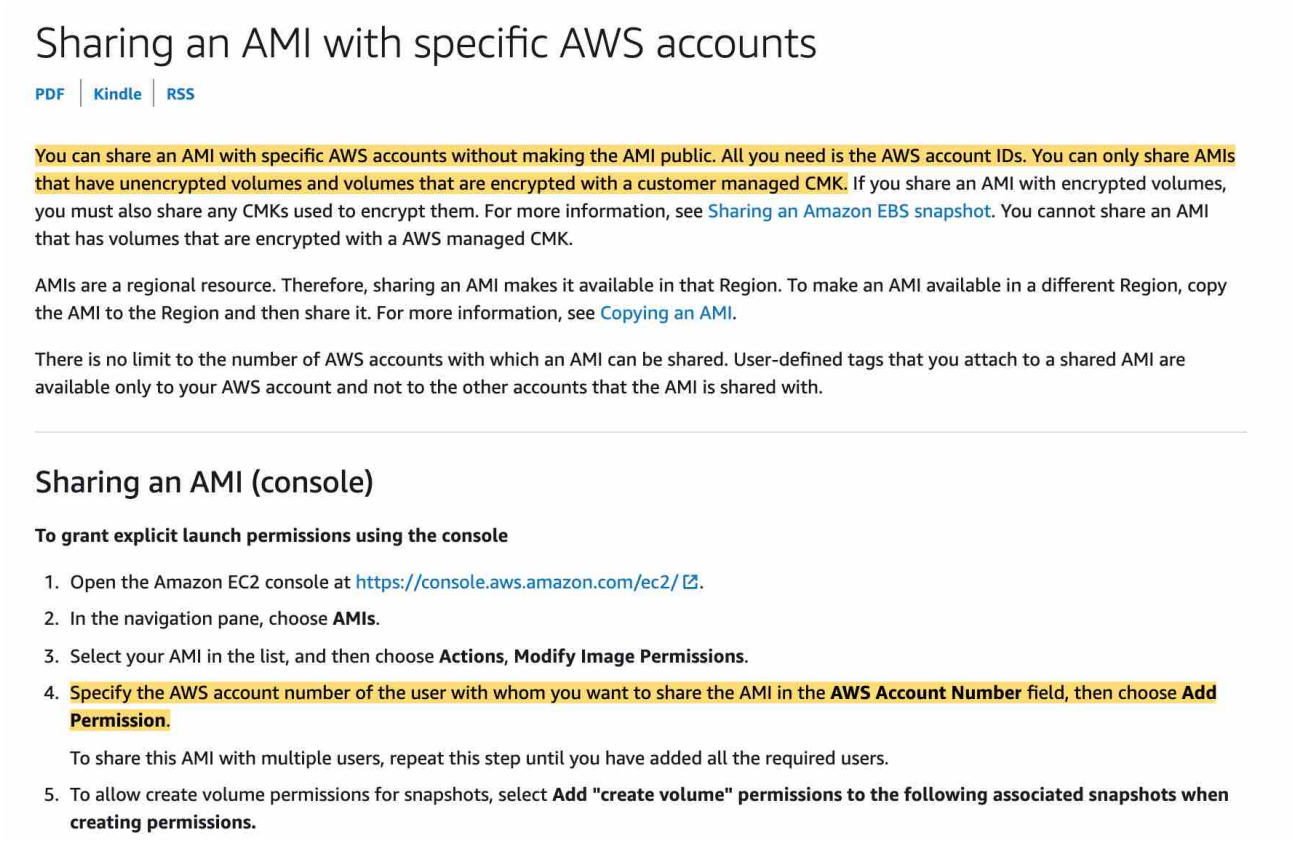
Correct option:

Edit the account list that can see the AMI from the AMI Console UI and the other accounts can start using it

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations. A shared AMI is an AMI that a developer created and made available for other developers to use.

You can share an AMI with specific AWS accounts without making the AMI public. All you need is the AWS account IDs. You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer-managed CMK. If you share an AMI with encrypted volumes, you must also share any CMKs used to encrypt them.

For the given use-case, you can modify image permissions for the AMI and then specify the AWS account number of the user with whom you want to share the AMI.



Sharing an AMI with specific AWS accounts

PDF | Kindle | RSS

You can share an AMI with specific AWS accounts without making the AMI public. All you need is the AWS account IDs. You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer managed CMK. If you share an AMI with encrypted volumes, you must also share any CMKs used to encrypt them. For more information, see Sharing an Amazon EBS snapshot. You cannot share an AMI that has volumes that are encrypted with a AWS managed CMK.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that Region. To make an AMI available in a different Region, copy the AMI to the Region and then share it. For more information, see Copying an AMI.

There is no limit to the number of AWS accounts with which an AMI can be shared. User-defined tags that you attach to a shared AMI are available only to your AWS account and not to the other accounts that the AMI is shared with.

Sharing an AMI (console)

To grant explicit launch permissions using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI in the list, and then choose **Actions**, **Modify Image Permissions**.
4. Specify the AWS account number of the user with whom you want to share the AMI in the **AWS Account Number** field, then choose **Add Permission**.

   To share this AMI with multiple users, repeat this step until you have added all the required users.
5. To allow create volume permissions for snapshots, select **Add "create volume" permissions to the following associated snapshots when creating permissions.**

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

Incorrect options:

The AMI can be shared without doing anything special. Just provide the target account with your secret AMI id and they can start using it

Edit the account list that can see the AMI from the AMI Console UI, and create an IAM role to be assumed by the other account using STS and the other accounts can start using it

Create an IAM role to be assumed by the other account using STS and they can start accessing your AMI

The correct process is described in the explanation above. These three options contradict the given explanation, therefore these are incorrect.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sharingamis-explicit.html

Question 25: **Correct**

VPC Peering has been enabled between VPC A and VPC B, and the route tables have been updated for VPC A. Still, your instances cannot communicate.

What is the most likely issue?

◯ **Check the NACL**

◯ **Check the route tables in VPC B** **(Correct)**

◯ **Check if DNS Resolution is enabled**

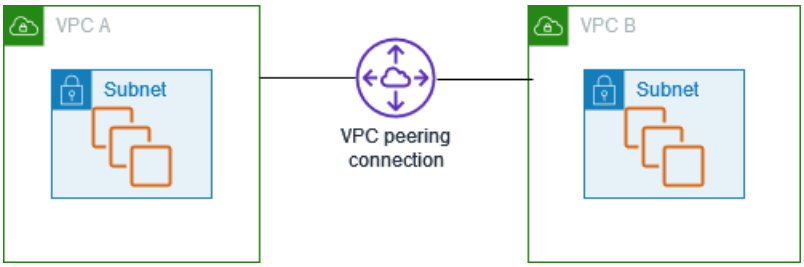
◯ **Check the instance security groups**

Explanation

Correct option:

Check the route tables in VPC B

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



via

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

You should note that to send private IPv4 traffic from your instance to an instance in a peer VPC, you must add a route to the route table that's associated with your subnet in which your instance resides. The owner of the peer VPC (VPC B in this case) must also complete these steps to add a route to direct traffic back to your VPC through the VPC peering connection, as usually this is the most likely cause for a failed communication between peered VPCs.

Incorrect options:

Check the NACL You should verify that an ALLOW rule exists in the network access control (network ACL) table for the required traffic.

Check the instance security groups - You should verify that the security group rules allow network traffic between the peered VPCs.

Check if DNS Resolution is enabled - DNS Resolution is used to enable resolution of public DNS hostnames to private IP addresses when queried from the peered VPC. This functionality also supports cross-account VPC peering so the two VPCs can be in different accounts.

Reference:

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

Question 26: **Correct**

You have tape backup processes and you would like to start migrating to the cloud to leverage the S3 storage capacity while keeping the same processes and iSCSI-compatible backup software you purchased a 10-year license for.

What do you recommend your company should be using?

- Volume Gateway
- File Gateway
- Tape Gateway (Correct)
- Snowball

Explanation

Correct option:

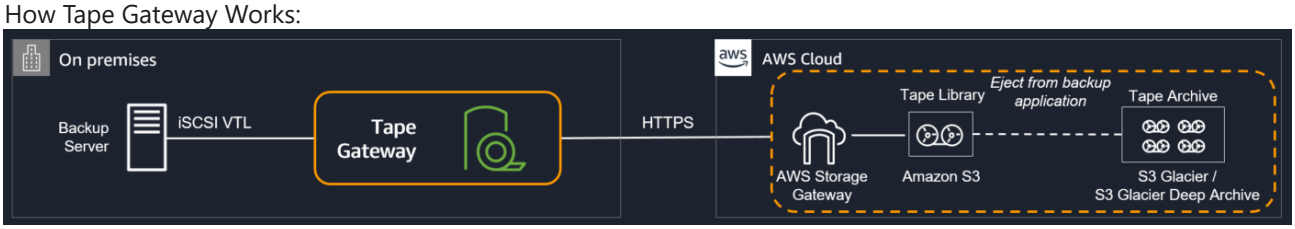
Tape Gateway

Tape Gateway enables you to replace using physical tapes on-premises with virtual tapes in AWS without changing existing backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on-premises for low-latency data access. Tape Gateway encrypts data between the gateway and AWS for secure data transfer and compresses data and transitions virtual tapes between Amazon S3 and Amazon S3 Glacier, or Amazon S3 Glacier Deep Archive, to minimize storage costs.

How Storage Gateway Works:

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

How Tape Gateway Works:



via - https://aws.amazon.com/storagegateway/vtl/

Incorrect options:

File Gateway - File Gateway provides a seamless way to connect to the cloud in order to store application data files and backup images as durable objects in Amazon S3 cloud storage. File Gateway offers SMB or NFS-based access to data in Amazon S3 with local caching. It can be used for on-premises applications, and for Amazon EC2-based applications that need file protocol access to S3 object storage.

File Gateway cannot be used to facilitate tape backup processes.

Volume Gateway - You can configure the AWS Storage Gateway service as a Volume Gateway to present cloud-based iSCSI block storage volumes to your on-premises applications. The Volume Gateway provides either a local cache or full volumes on-premises while also storing full copies of your volumes in the AWS cloud. Volume Gateway also provides Amazon EBS Snapshots of your data for backup, disaster recovery, and migration. It's easy to get started with the Volume Gateway: Deploy it as a virtual machine or hardware appliance, give it local disk resources, connect it to your applications, and start using your hybrid cloud storage for block data.

Volume Gateway cannot be used to facilitate tape backup processes.

Snowball - AWS Snowball, a part of the AWS Snow Family, is a data migration and edge computing device that comes in two options. Snowball Edge Storage Optimized devices provide both block storage and Amazon S3-compatible object storage, and 40 vCPUs. They are well suited for local storage and large scale data transfer. Snowball Edge Compute Optimized devices provide 52 vCPUs, block and object storage, and an optional GPU for use cases like advanced machine learning and full-motion video analysis in disconnected environments.

Snowball Edge Storage Optimized is the optimal choice if you need to securely and quickly transfer dozens of terabytes to petabytes of data to AWS. It provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity to address large scale data transfer and pre-processing use cases.

Snowball cannot be used to facilitate tape backup processes.

References:

https://aws.amazon.com/storagegateway/vtl/

https://aws.amazon.com/storagegateway/

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

Question 27: **Correct**

An EC2 instance, which is part of an Auto Scaling Group (ASG), has been marked as unhealthy because of a health check.

What is the outcome of the instance being marked as unhealthy?

- ○ **If a custom health check marks the instance as unhealthy, then ASG will not replace the unhealthy instance automatically**

- ○ **The health check status has to be defined as unhealthy by both EC2 instance and Elastic Load Balancer for the ASG to replace the instance**

○ **ASG will replace the unhealthy instance with a healthy instance** **(Correct)**

○ **An instance can automatically recover its health in the specified grace period. Hence, ASG will wait for the grace period to expire before replacing the unhealthy instance**

Explanation

Correct option:

ASG will replace the unhealthy instance with a healthy instance - After an instance has been marked unhealthy because of a health check, it is almost immediately scheduled for replacement. It never automatically recovers its health. Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.

When your instance is terminated, any associated Elastic IP addresses are dissociated and are not automatically associated with the new instance. You must associate these Elastic IP addresses with the new instance manually.

Incorrect options:

An instance can automatically recover its health in the specified grace period. Hence, ASG will wait for the grace period to expire before replacing the unhealthy instance - An instance never automatically recovers its health. When an instance launches, Amazon EC2 Auto Scaling uses the value of the `HealthCheckGracePeriod` for the Auto Scaling group to determine how long to wait before checking the health status of the instance. By default, the health check grace period is 300 seconds when you create an Auto Scaling group from the AWS Management Console. Its default value is 0 seconds when you create an Auto Scaling group using the AWS CLI or an SDK.

The health check status has to be defined as unhealthy by both EC2 instance and Elastic Load Balancer for the ASG to replace the instance - ASG configuration can either consider EC2 health status checks or ELB health status checks, not both.

If a custom health check marks the instance as unhealthy, then ASG will not replace the unhealthy instance automatically - If you have custom health checks, you can send the information from your health checks to Amazon EC2 Auto Scaling so that Amazon EC2 Auto Scaling can use this information. For example, if you determine that an instance is not functioning as expected, you can set the health status of the instance to Unhealthy. The next time that Amazon EC2 Auto Scaling performs a health check on the instance, it will determine that the instance is unhealthy and then launch a replacement instance.

Reference:

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

Question 28: **Incorrect**

You are setting up a distributed in-memory database and you would like to auto-scale your Auto Scaling Group based on the average RAM usage of your EC2 instances.

How can you achieve this?

○ **Place the instances behind a load balancer, which will have the capability of monitoring the RAM of the EC2 instances**

~~with the smart balancing feature~~

- ○ **Auto Scale your ASG based on the CPUUtilization metric** **(Incorrect)**

- ○ **Enable EC2 detailed monitoring and use the CloudWatch metric RAMUtilization to setup scaling policies**

- ○ **Push the RAMUtilization as a custom metric using custom scripts in EC2 and setup scaling policies using this metric** **(Correct)**

Explanation

Correct option:

Push the RAMUtilization as a custom metric using custom scripts in EC2 and setup scaling policies using this metric

You can publish your own metrics to CloudWatch using the AWS CLI or an API. You can view statistical graphs of your published metrics with the AWS Management Console. Metrics produced by AWS services are standard resolution by default.

Each metric is one of the following:

Standard resolution, with data having a one-minute granularity

High resolution, with data at a granularity of one second

When you publish a custom metric, you can define it as either standard resolution or high resolution. When you publish a high-resolution metric, CloudWatch stores it with a resolution of 1 second, and you can read and retrieve it with a period of 1 second, 5 seconds, 10 seconds, 30 seconds, or any multiple of 60 seconds.

For example, the following command publishes a Buffers metric with two dimensions named InstanceId and InstanceType: `aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small`

For the given use-case, you can set up RAMUtilization as a custom metric and push it to CloudWatch to be further used in the scaling policy.

Incorrect options:

Enable EC2 detailed monitoring and use the CloudWatch metric RAMUtilization to setup scaling policies - RAMUtilization is not available as an EC2 metric out-of-the-box, so this option is incorrect.

Auto Scale your ASG based on the CPUUtilization metric - The use-case refers to RAM usage as the relevant metric, so this option is not correct.

Place the instances behind a load balancer, which will have the capability of monitoring the RAM of the EC2 instances with the smart balancing feature - This option has been added as a distractor. There is no such thing as monitoring the RAM of the EC2 instance via the smart balancing feature.

Reference:

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

Question 29: **Correct**

Your e-commerce website has a few really popular items that constitute 10% of your portfolio items in your RDS database but represent 90% of your traffic. Your database is starting to struggle with the read demand and your CTO tasked you with designing a solution to improve the read scalability on the database side.

What do you recommend? (Select two)

- [ ] **Setup a Multi-AZ RDS database**
- [x] **Setup an ElastiCache cluster** **(Correct)**
- [ ] **Setup an API gateway with cache enabled in front of your database**
- [ ] **Setup a DAX cluster**

Setup Read Replicas (Correct)

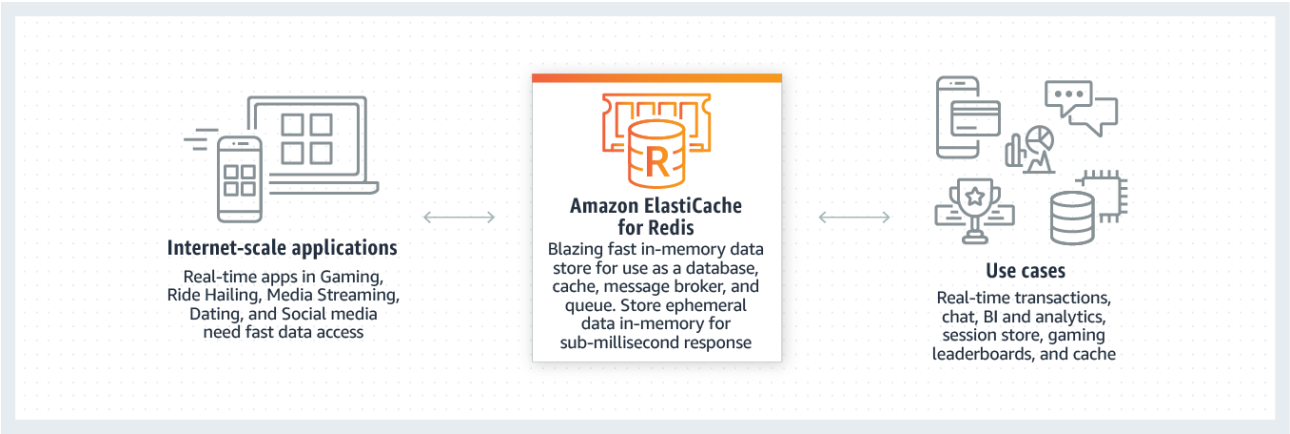
Explanation

Correct options:

Setup an ElastiCache cluster

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

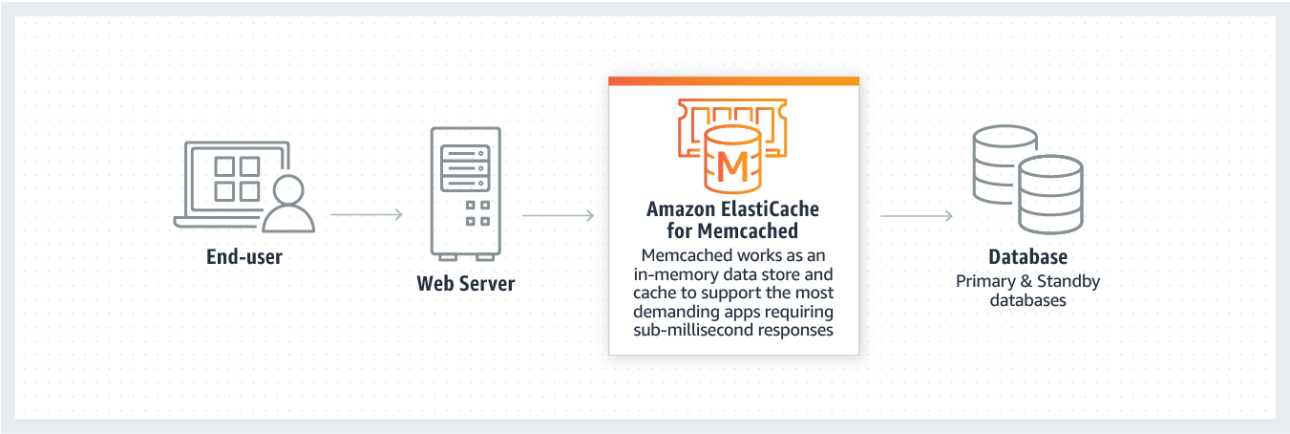
How ElastiCache for Redis works:



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

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

How ElastiCache for Memcached works:



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

For the given use-case, you can use ElastiCache in front of the RDS database to improve the read scalability.

Exam Alert:

Please review this comparison sheet for Redis vs Memcached features:

Choosing between Redis and Memcached

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

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

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

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

Setup Read Replicas

Amazon RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances. They make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. For the MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance. It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance.

Overview of Amazon RDS read replicas

Deploying one or more read replicas for a given source DB instance might make sense in a variety of scenarios, including the following:

- Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads. You can direct this excess read traffic to one or more read replicas.
- Serving read traffic while the source DB instance is unavailable. In some cases, your source DB instance might not be able to take I/O requests, for example due to I/O suspension for backups or scheduled maintenance. In these cases, you can direct read traffic to your read replicas. For this use case, keep in mind that the data on the read replica might be "stale" because the source DB instance is unavailable.
- Business reporting or data warehousing scenarios where you might want business reporting queries to run against a read replica, rather than your production DB instance.
- Implementing disaster recovery. You can promote a read replica to a standalone instance as a disaster recovery solution if the primary DB instance fails.

By default, a read replica is created with the same storage type as the source DB instance. However, you can create a read replica that has a different storage type from the source DB instance based on the options listed in the following table.

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

Incorrect options:

Setup a Multi-AZ RDS database - Amazon RDS Multi-AZ deployments provide enhanced availability and durability for RDS database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). You cannot use Multi-AZ to enhance the read performance for the RDS databases.

Setup a DAX cluster - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-master, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DAX is a DynamoDB-compatible caching service that enables you to benefit from fast in-memory performance for demanding applications. DAX cannot be used as a caching layer for a relational database.

Setup an API gateway with cache enabled in front of your database - This is a distractor as an API Gateway with the cache enabled is typically used to front the web servers based on EC2 instances or even Lambda based serverless solutions.

References:

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

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

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

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

Question 30: **Correct**

You operate a technology company that implements the Netflix chaos testing in production. This means that your EC2 instances in production can be terminated at any time, to test the resiliency of your applications. You have been experiencing a lot of 4XXs errors lately on your website that is exposed by a load balancer, and you realize you cannot SSH into the instances that were producing these errors as they have been terminated.

How can you gain access to logs files that describe the list of HTTP requests that were inducing these problems?

- ○ **Look at the EC2 default logs in CloudWatch Logs**
- ○ **Contact AWS Support to recover the instances**
- ○ **Use EC2 Rescue and bring back the log files from the wiped EBS volumes**
- ○ **Enable the ELB access logs and query them using Athena** **(Correct)**

Explanation

Correct option:

Enable the ELB access logs and query them using Athena

ELB access logs is an optional feature of Elastic Load Balancing that is disabled by default. The access logs capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues. Each access log file is automatically encrypted using SSE-S3 before it is stored in your S3 bucket and decrypted when you access it. You do not need to take any action; the encryption and decryption is performed transparently.

Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries.

For the given use-case, you can enable ELB access logs and then use Athena to analyze the 4XX errors from the log files stored in S3.

Access logs for your Application Load Balancer

PDF | Kindle | RSS

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues.

Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Load Balancing captures the logs and stores them in the Amazon S3 bucket that you specify as compressed files. You can disable access logging at any time.

Each access log file is automatically encrypted using SSE-S3 before it is stored in your S3 bucket and decrypted when you access it. You do not need to take any action; the encryption and decryption is performed transparently. Each log file is encrypted with a unique key, which is itself encrypted with a master key that is regularly rotated. For more information, see Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3) in the *Amazon Simple Storage Service Developer Guide*.

There is no additional charge for access logs. You are charged storage costs for Amazon S3, but not charged for the bandwidth used by Elastic Load Balancing to send log files to Amazon S3. For more information about storage costs, see Amazon S3 pricing.

via - https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html

Incorrect options:

Use EC2 Rescue and bring back the log files from the wiped EBS volumes - EC2Rescue for Linux is an easy-to-use, open-source tool that can be run on an Amazon EC2 Linux instance to diagnose and troubleshoot common issues using its library of over 100 modules. A few generalized use cases for EC2Rescue for Linux include gathering syslog and package manager logs, collecting resource utilization data, and diagnosing/remediating known problematic kernel parameters and common OpenSSH issues.

Since the use-case mentions that the instances have been terminated, so this tool cannot be used for such analysis.

Contact AWS Support to recover the instances - You cannot recover terminated EC2 instances.

Look at the EC2 default logs in CloudWatch Logs There are no default logs for EC2 in CloudWatch Logs. You need to set up the CloudWatch Agent to collect logs from Amazon EC2 instances and on-premises servers,

running either Linux or Windows Server.

Reference:

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Linux-Server-EC2Rescue.html

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html

Question 31: **Correct**

Your infrastructure runs a daily job to compute different metrics based on all the resources that are running in your account. The goal of this job is to provide you with metrics that will be pushed into a reporting Tableau dashboard and allow your SysOps Administrator to make good decisions to bring the cost down. That job is fault-tolerant and can be resumed at any time.

Which EC2 instance type would you choose to keep costs low?

- ◯ **EC2 Spot Instances** **(Correct)**
- ◯ **EC2 Reserved Instances**
- ◯ **EC2 Placement Groups - Cluster**
- ◯ **EC2 On Demand**

Explanation

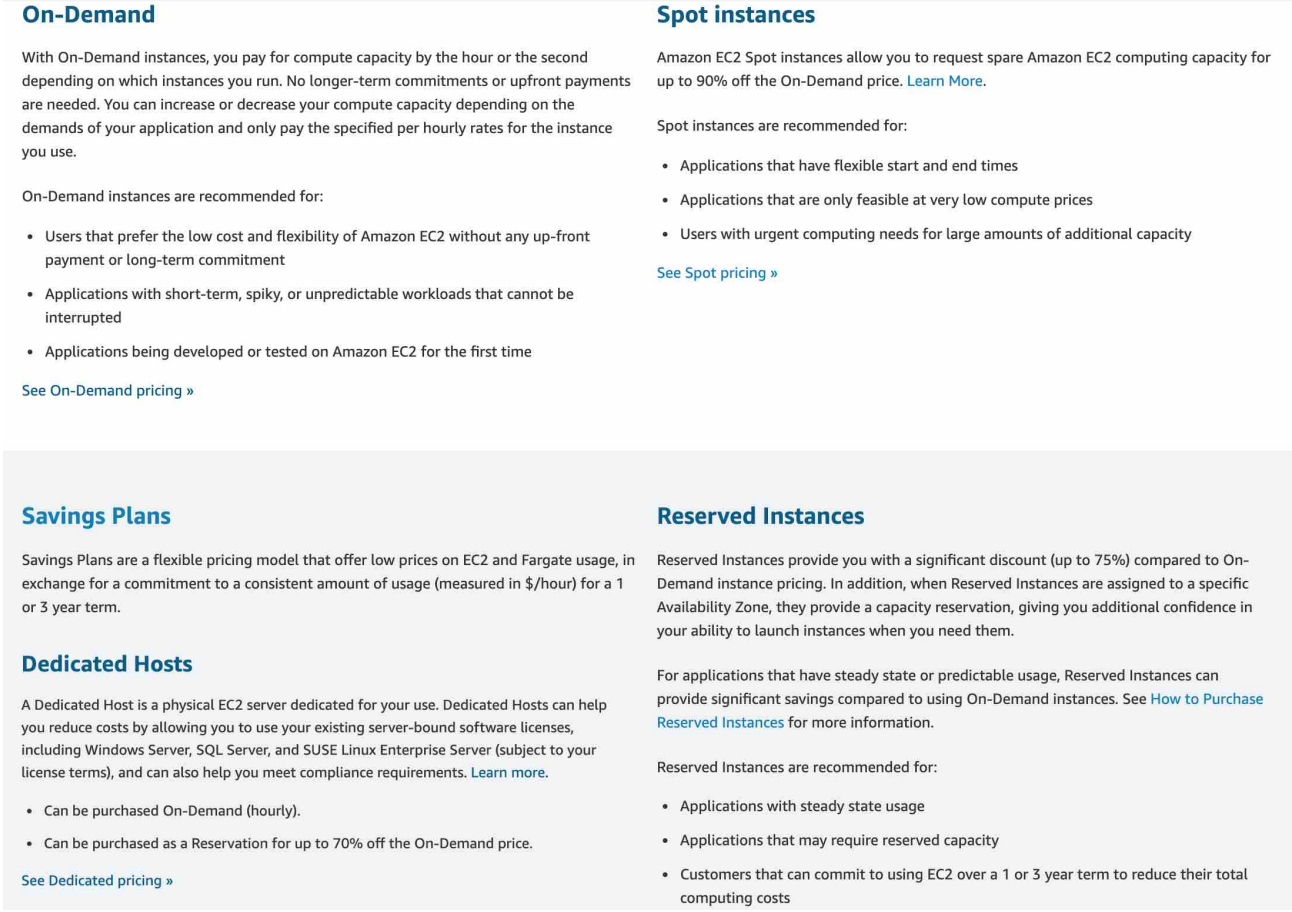
Correct option:

EC2 Spot Instances

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price (up to 90% off the On-Demand price). Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2 and adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available and the maximum price per hour for your request exceeds the Spot price.

Since the job is fault-tolerant and can be resumed at any time, therefore Spot Instances are a good fit for the given use-case.

Please see this detailed overview of various types of EC2 instances from a pricing perspective:



On-Demand

With On-Demand instances, you pay for compute capacity by the hour or the second depending on which instances you run. No longer-term commitments or upfront payments are needed. You can increase or decrease your compute capacity depending on the demands of your application and only pay the specified per hourly rates for the instance you use.

On-Demand instances are recommended for:

- Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on Amazon EC2 for the first time

See On-Demand pricing »

Spot instances

Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price. Learn More.

Spot instances are recommended for:

- Applications that have flexible start and end times
- Applications that are only feasible at very low compute prices
- Users with urgent computing needs for large amounts of additional capacity

See Spot pricing »

Savings Plans

Savings Plans are a flexible pricing model that offer low prices on EC2 and Fargate usage, in exchange for a commitment to a consistent amount of usage (measured in $/hour) for a 1 or 3 year term.

Dedicated Hosts

A Dedicated Host is a physical EC2 server dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses, including Windows Server, SQL Server, and SUSE Linux Enterprise Server (subject to your license terms), and can also help you meet compliance requirements. Learn more.

- Can be purchased On-Demand (hourly).
- Can be purchased as a Reservation for up to 70% off the On-Demand price.

See Dedicated pricing »

Reserved Instances

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

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

Reserved Instances are recommended for:

- Applications with steady state usage
- Applications that may require reserved capacity
- Customers that can commit to using EC2 over a 1 or 3 year term to reduce their total computing costs

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

Incorrect options:

EC2 On-Demand - With On-Demand Instances, you pay for compute capacity by the second with no long-term commitments. You have full control over its lifecycle—you decide when to launch, stop, hibernate, start, reboot, or terminate it. There is no long-term commitment required when you purchase On-Demand Instances. You pay only for the seconds that your On-Demand Instances are running. AWS recommends that you use On-Demand Instances for applications with short-term, irregular workloads that cannot be interrupted. On-Demand Instances would be costlier compared to Spot Instances for the given use-case.

EC2 Reserved Instances - Reserved instances reduce your Amazon EC2 costs by making a commitment to a consistent instance configuration, including instance type and Region, for a term of 1 or 3 years. For the given use case, this kind of annual commitment might not be a desirable option, as the daily job runs just for some given duration in a day which is better served via the Spot Instances.

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

Cluster – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.

Partition – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.

Spread – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

This option has been added as a distractor as the cluster placement group would have no bearing on reducing the cost of the solution.

Reference:

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

Question 32: **Correct**

You are launching an EC2 instance and it fails with an `InsufficientInstanceCapacity` error. What should you do?

- ◯ **Try to launch the instance in another AZ** **(Correct)**
- ◯ **Request for a service limit increase in AWS support console**
- ◯ **Run Amazon Inspector on your EC2 instances to find out what's consuming the capacity**
- ◯ **Use AWS Trusted Advisor to understand the root cause of this issue**

Explanation

Correct option:

Try to launch the instance in another AZ

You get the InsufficientInstanceCapacity error when you try to launch a new instance or restart a stopped instance. `InsufficientInstanceCapacity` error implies that AWS does not have the capacity to serve your request.

Exam Alert:

Please make sure that you understand the differences between the `InsufficientInstanceCapacity` error and the `InstanceLimitExceeded` error as these are commonly probed in the exam.

Troubleshooting instance launch issues:

Instance limit exceeded

Description

You get the `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get an `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance, you have reached the limit on the number of instances that you can launch in a Region. When you create your AWS account, we set default limits on the number of instances you can run on a per-Region basis.

Solution

You can request an instance limit increase on a per-region basis. For more information, see Amazon EC2 service quotas.

Insufficient instance capacity

Description

You get the `InsufficientInstanceCapacity` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get this error when you try to launch an instance or restart a stopped instance, AWS does not currently have enough available On-Demand capacity to fulfill your request.

Solution

To resolve the issue, try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- If you're launching an instance, submit a new request without specifying an Availability Zone.
- If you're launching an instance, submit a new request using a different instance type (which you can resize at a later stage). For more information, see Changing the instance type.
- If you are launching instances into a cluster placement group, you can get an insufficient capacity error. For more information, see Placement group rules and limitations.

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

Incorrect options:

Run Amazon Inspector on your EC2 instances to find out what's consuming the capacity

Use AWS Trusted Advisor to understand the root cause of this issue

Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances.

AWS Trusted Advisor is an online tool that provides you real-time guidance to help you provision your resources following AWS best practices. Whether establishing new workflows, developing applications, or as part of ongoing improvement, you can take advantage of the recommendations provided by Trusted Advisor regularly to help keep your solutions provisioned optimally.

Both these options have been added as distractors as neither Amazon Inspector nor AWS Trusted Advisor can help in solving the `InsufficientInstanceCapacity` error.

Request for a service limit increase in AWS support console - There is no need to request for a service limit increase to handle the `InsufficientInstanceCapacity` error. Please refer to the solutions described in the explanation above.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/troubleshooting-launch.html

Question 33: **Incorrect**

The security team at your travel company has detected a series of malicious attacks on port 846. As such, it needs to ensure that all your security groups are compliant with having this port closed, at all times. In the event such a port is being opened, you need to receive a notification as soon as possible.

Which service can help you with achieving such task?

- AWS GuardDuty
- AWS WAF **(Incorrect)**
- AWS Shield
- AWS Config **(Correct)**

Explanation

Correct option:

AWS Config

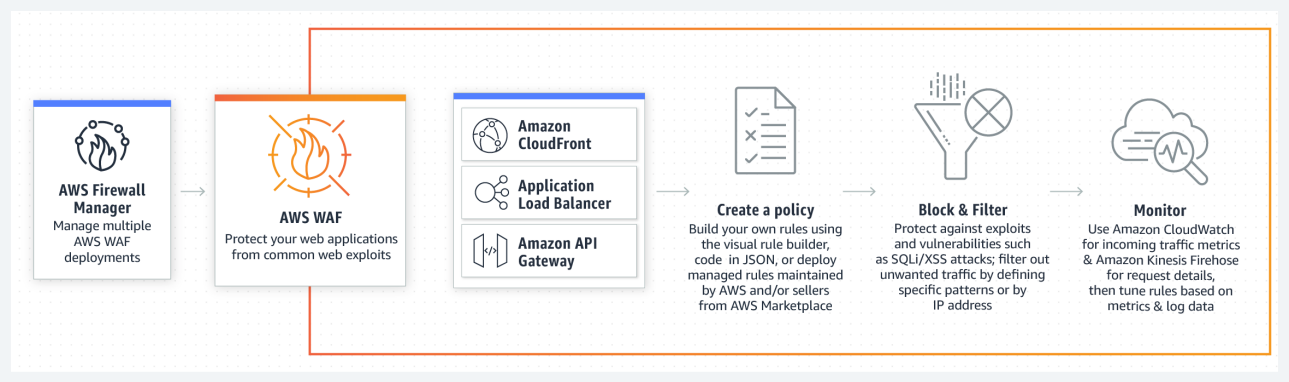
AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?".

You can use an EventBridge rule with a custom event pattern and an input transformer to match an AWS Config evaluation rule output as NON_COMPLIANT. Then, route the response to an Amazon Simple Notification Service (Amazon SNS) topic.

Incorrect options:

AWS WAF - AWS WAF is a web application firewall that helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define. These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection, and cross-site scripting. WAF cannot be used to detect and get notified about any security gaps when port 846 is opened.

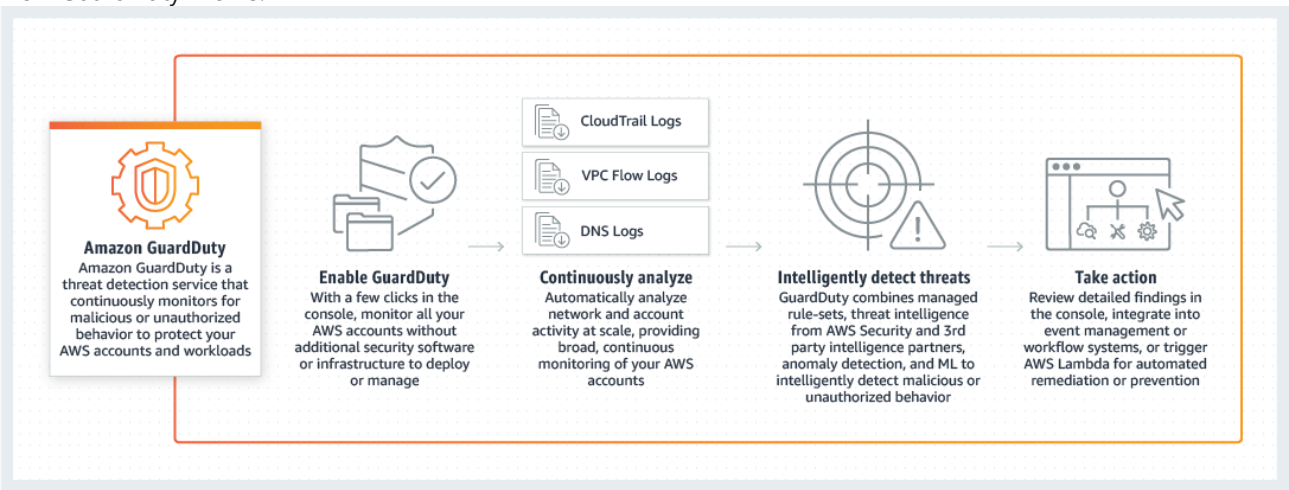
How WAF Works:



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

AWS GuardDuty - - GuardDuty is a threat detection service that monitors malicious activity and unauthorized behavior to protect your AWS account. GuardDuty analyzes billions of events across your AWS accounts from AWS CloudTrail (AWS user and API activity in your accounts), Amazon VPC Flow Logs (network traffic data), and DNS Logs (name query patterns). GuardDuty cannot be used to detect and get notified about any security gaps when port 846 is opened.

How GuardDuty Works:



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

AWS Shield - AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS. AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection. AWS Shield cannot be used to detect and get notified about any security gaps when port 846 is opened.

References:

https://aws.amazon.com/config/

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

https://aws.amazon.com/shield/

Question 34: **Correct**

One of your web applications runs behind a load balancer and an auto scaling group, which has a scaling policy based on the backend Aurora database requests. On top of scaling the ASG, the CloudWatch alarms auto scale the Aurora database. After a few scale out and scale in events, your application has completely lost connectivity to the database. You check the database URL referenced in the SSM parameter store and it turns out that it does not correspond to any of the Aurora read replicas, although it used to.

How can you fix that problem easily in the long term while allowing your application to remain elastic?

- ○ **Create a target group made up of the Aurora Read Replicas and set up a Network Load Balancer**
- ○ **Use the Aurora Reader Endpoint** **(Correct)**
- ○ **Disable Aurora Auto Scaling**
- ○ **Create an AWS Lambda function CRON job that updates SSM with the latest connection string from all the alive Aurora Read Replicas**

Explanation

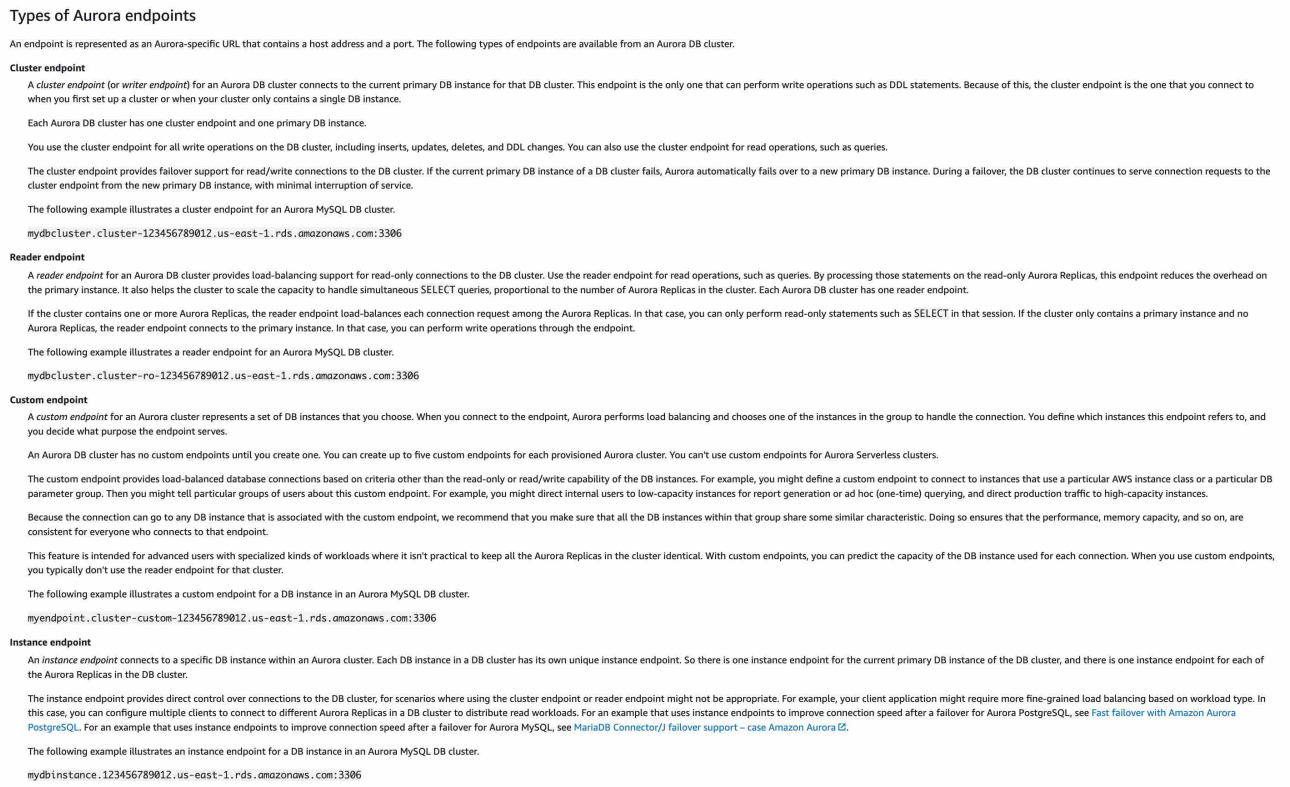
Correct option:

Use the Aurora Reader Endpoint

To perform queries, you can connect to the reader endpoint, with Aurora automatically performing load-balancing among all the Aurora Replicas.

A reader endpoint for an Aurora DB cluster provides load-balancing support for read-only connections to the DB cluster. Use the reader endpoint for read operations, such as queries. By processing those statements on the read-only Aurora Replicas, this endpoint reduces the overhead on the primary instance. It also helps the cluster to scale the capacity to handle simultaneous SELECT queries, proportional to the number of Aurora Replicas in the cluster. Each Aurora DB cluster has one reader endpoint.

If the cluster contains one or more Aurora Replicas, the reader endpoint load-balances each connection request among the Aurora Replicas. In that case, you can only perform read-only statements such as SELECT in that session.



Types of Aurora endpoints

An endpoint is represented as an Aurora-specific URL that contains a host address and a port. The following types of endpoints are available from an Aurora DB cluster.

Cluster endpoint

A *cluster endpoint* (or *writer endpoint*) for an Aurora DB cluster connects to the current primary DB instance for that DB cluster. This endpoint is the only one that can perform write operations such as DDL statements. Because of this, the cluster endpoint is the one that you connect to when you first set up a cluster or when your cluster only contains a single DB instance.

Each Aurora DB cluster has one cluster endpoint and one primary DB instance.

You use the cluster endpoint for all write operations on the DB cluster, including inserts, updates, deletes, and DDL changes. You can also use the cluster endpoint for read operations, such as queries.

The cluster endpoint provides failover support for read/write connections to the DB cluster. If the current primary DB instance of a DB cluster fails, Aurora automatically fails over to a new primary DB instance. During a failover, the DB cluster continues to serve connection requests to the cluster endpoint from the new primary DB instance, with minimal interruption of service.

The following example illustrates a cluster endpoint for an Aurora MySQL DB cluster.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306
```

Reader endpoint

A *reader endpoint* for an Aurora DB cluster provides load-balancing support for read-only connections to the DB cluster. Use the reader endpoint for read operations, such as queries. By processing those statements on the read-only Aurora Replicas, this endpoint reduces the overhead on the primary instance. It also helps the cluster to scale the capacity to handle simultaneous SELECT queries, proportional to the number of Aurora Replicas in the cluster. Each Aurora DB cluster has one reader endpoint.

If the cluster contains one or more Aurora Replicas, the reader endpoint load-balances each connection request among the Aurora Replicas. In that case, you can only perform read-only statements such as SELECT in that session. If the cluster only contains a primary instance and no Aurora Replicas, the reader endpoint connects to the primary instance. In that case, you can perform write operations through the endpoint.

The following example illustrates a reader endpoint for an Aurora MySQL DB cluster.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306
```

Custom endpoint

A *custom endpoint* for an Aurora cluster represents a set of DB instances that you choose. When you connect to the endpoint, Aurora performs load balancing and chooses one of the instances in the group to handle the connection. You define which instances this endpoint refers to, and you decide what purpose the endpoint serves.

An Aurora DB cluster has no custom endpoints until you create one. You can create up to five custom endpoints for each provisioned Aurora cluster. You can't use custom endpoints for Aurora Serverless clusters.

The custom endpoint provides load-balanced database connections based on criteria other than the read-only or read/write capability of the DB instances. For example, you might define a custom endpoint to connect to instances that use a particular AWS instance class or a particular DB parameter group. Then you might tell particular groups of users about this custom endpoint. For example, you might direct internal users to low-capacity instances for report generation or ad hoc (one-time) querying, and direct production traffic to high-capacity instances.

Because the connection can go to any DB instance that is associated with the custom endpoint, we recommend that you make sure that all the DB instances within that group share some similar characteristic. Doing so ensures that the performance, memory capacity, and so on, are consistent for everyone who connects to that endpoint.

This feature is intended for advanced users with specialized kinds of workloads where it isn't practical to keep all the Aurora Replicas in the cluster identical. With custom endpoints, you can predict the capacity of the DB instance used for each connection. When you use custom endpoints, you typically don't use the reader endpoint for that cluster.

The following example illustrates a custom endpoint for a DB instance in an Aurora MySQL DB cluster.

```
myendpoint.cluster-custom-123456789012.us-east-1.rds.amazonaws.com:3306
```

Instance endpoint

An *instance endpoint* connects to a specific DB instance within an Aurora cluster. Each DB instance in a DB cluster has its own unique instance endpoint. So there is one instance endpoint for the current primary DB instance of the DB cluster, and there is one instance endpoint for each of the Aurora Replicas in the DB cluster.

The instance endpoint provides direct control over connections to the DB cluster, for scenarios where using the cluster endpoint or reader endpoint might not be appropriate. For example, your client application might require more fine-grained load balancing based on workload type. In this case, you can configure multiple clients to connect to different Aurora Replicas in a DB cluster to distribute read workloads. For an example that uses instance endpoints to improve connection speed after a failover for Aurora PostgreSQL, see Fast failover with Amazon Aurora PostgreSQL. For an example that uses instance endpoints to improve connection speed after a failover for Aurora MySQL, see MariaDB Connector/J failover support – case Amazon Aurora.

The following example illustrates an instance endpoint for a DB instance in an Aurora MySQL DB cluster.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com:3306
```

via - https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.Endpoints.html

Incorrect options:

Disable Aurora Auto Scaling - This option has been added as a distractor as you cannot disable Aurora Auto Scaling.

Create an AWS Lambda function CRON job that updates SSM with the latest connection string from all the alive Aurora Read Replicas - The Lambda CRON job can update the SSM with the connection string for Read Replicas but it is neither a simple nor an efficient solution.

Create a target group made up of the Aurora Read Replicas and set up a Network Load Balancer - Amazon Aurora typically involves a cluster of DB instances instead of a single instance. Each connection is handled by a specific DB instance. When you connect to an Aurora cluster, the hostname and port point to an intermediate handler called an endpoint. Aurora uses the endpoint mechanism to abstract these connections. Thus, you don't have to hardcode all the hostnames or write your own logic for load-balancing and rerouting connections when some DB instances aren't available. You should note that the Elastic Load Balancer cannot be used to load balance against RDS databases.

Reference:

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

Question 35: **Correct**

You would like to ensure that your instances in your private subnet for us-west-1b can talk to your public instances in us-west-1c using their private IP addresses.

How can you establish network connectivity between the two subnets in the simplest possible way?

◯ **Create one VPC with two subnets** **(Correct)**

◯ **Use a NAT Gateway**

◯ **Use a NAT instance**

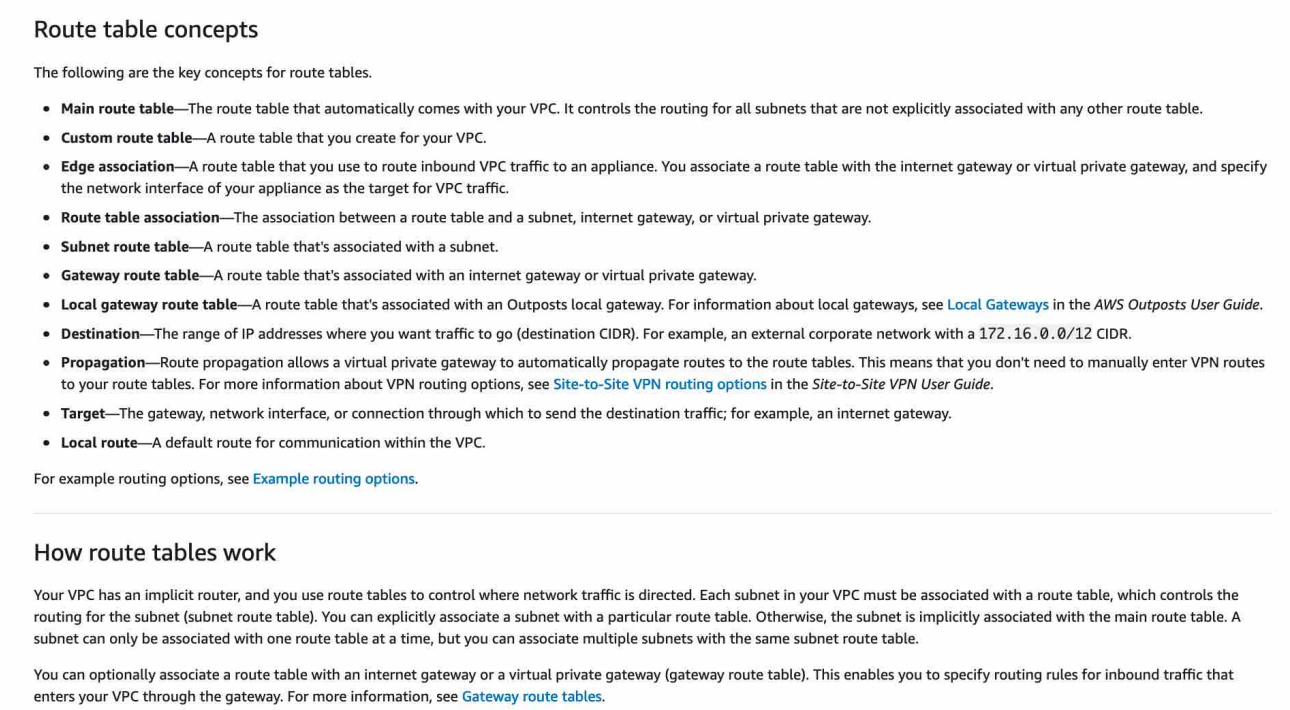
◯ **Create two VPCs with one subnet each, and peer them**

Explanation

Correct option:

Create one VPC with two subnets - When you create a new VPC, there is a main route table that automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table. Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet (subnet route table). You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same subnet route table.

So for the given use-case, as the VPC has two subnets, the main route table will take care of the routing between the two subnets.



Route table concepts

The following are the key concepts for route tables.

- **Main route table**—The route table that automatically comes with your VPC. It controls the routing for all subnets that are not explicitly associated with any other route table.
- **Custom route table**—A route table that you create for your VPC.
- **Edge association**—A route table that you use to route inbound VPC traffic to an appliance. You associate a route table with the internet gateway or virtual private gateway, and specify the network interface of your appliance as the target for VPC traffic.
- **Route table association**—The association between a route table and a subnet, internet gateway, or virtual private gateway.
- **Subnet route table**—A route table that's associated with a subnet.
- **Gateway route table**—A route table that's associated with an internet gateway or virtual private gateway.
- **Local gateway route table**—A route table that's associated with an Outposts local gateway. For information about local gateways, see Local Gateways in the *AWS Outposts User Guide*.
- **Destination**—The range of IP addresses where you want traffic to go (destination CIDR). For example, an external corporate network with a `172.16.0.0/12` CIDR.
- **Propagation**—Route propagation allows a virtual private gateway to automatically propagate routes to the route tables. This means that you don't need to manually enter VPN routes to your route tables. For more information about VPN routing options, see Site-to-Site VPN routing options in the *Site-to-Site VPN User Guide*.
- **Target**—The gateway, network interface, or connection through which to send the destination traffic; for example, an internet gateway.
- **Local route**—A default route for communication within the VPC.

For example routing options, see Example routing options.

How route tables work

Your VPC has an implicit router, and you use route tables to control where network traffic is directed. Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet (subnet route table). You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same subnet route table.

You can optionally associate a route table with an internet gateway or a virtual private gateway (gateway route table). This enables you to specify routing rules for inbound traffic that enters your VPC through the gateway. For more information, see Gateway route tables.

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

Incorrect options:

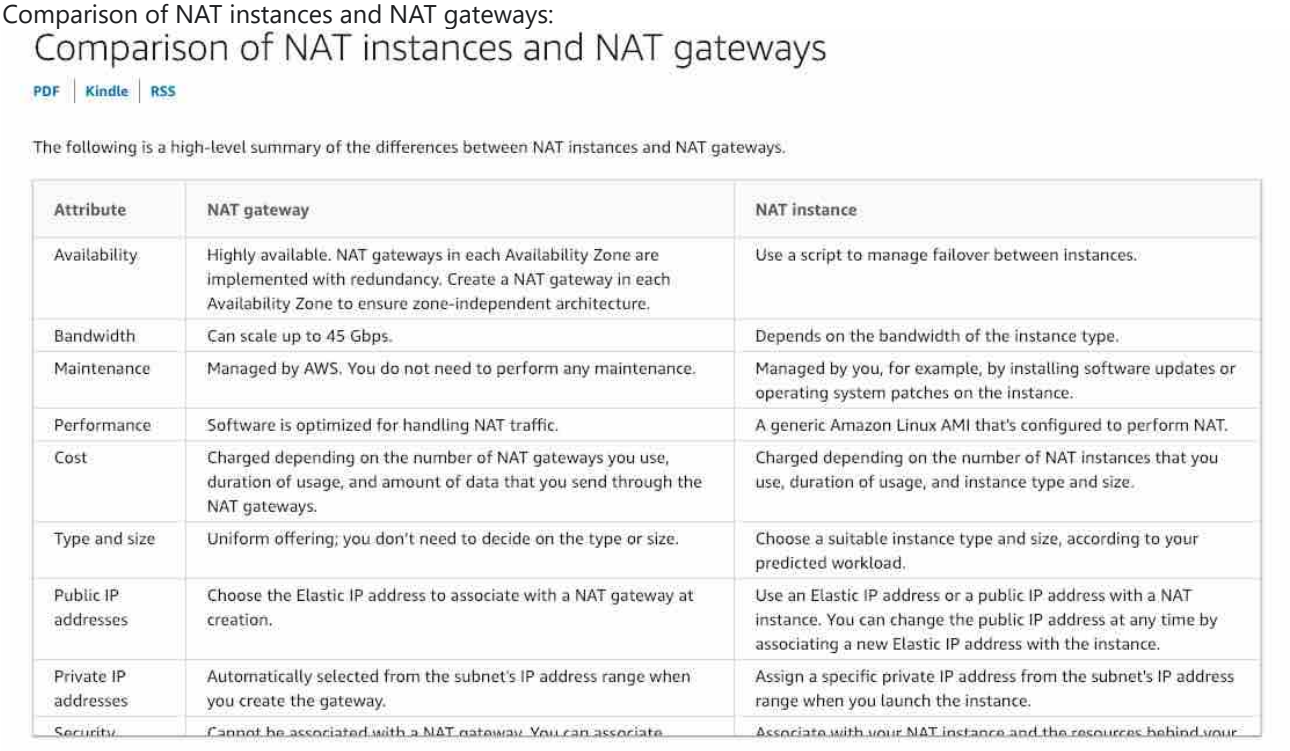
Create two VPCs with one subnet each, and peer them - A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same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

Use a NAT Gateway - You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

Use a NAT instance - You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the internet or other AWS services, but

prevent the instances from receiving inbound traffic initiated by someone on the internet.

Comparison of NAT instances and NAT gateways:



Comparison of NAT instances and NAT gateways

PDF | Kindle | RSS

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

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

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

References:

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

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

Question 36: **Correct**

Your application has complex runtime and OS dependencies and is taking a long time to be deployed on Elastic Beanstalk. You cannot sacrifice application availability.

What should you do to improve the deployment time? (Select two)

Use rolling with additional batch

Create a new beanstalk environment for each application and apply blue/green deployment patterns **(Correct)**

Use all at once deployment pattern

Upgrade the EC2 instance type

Create a Golden AMI with your application (Correct)

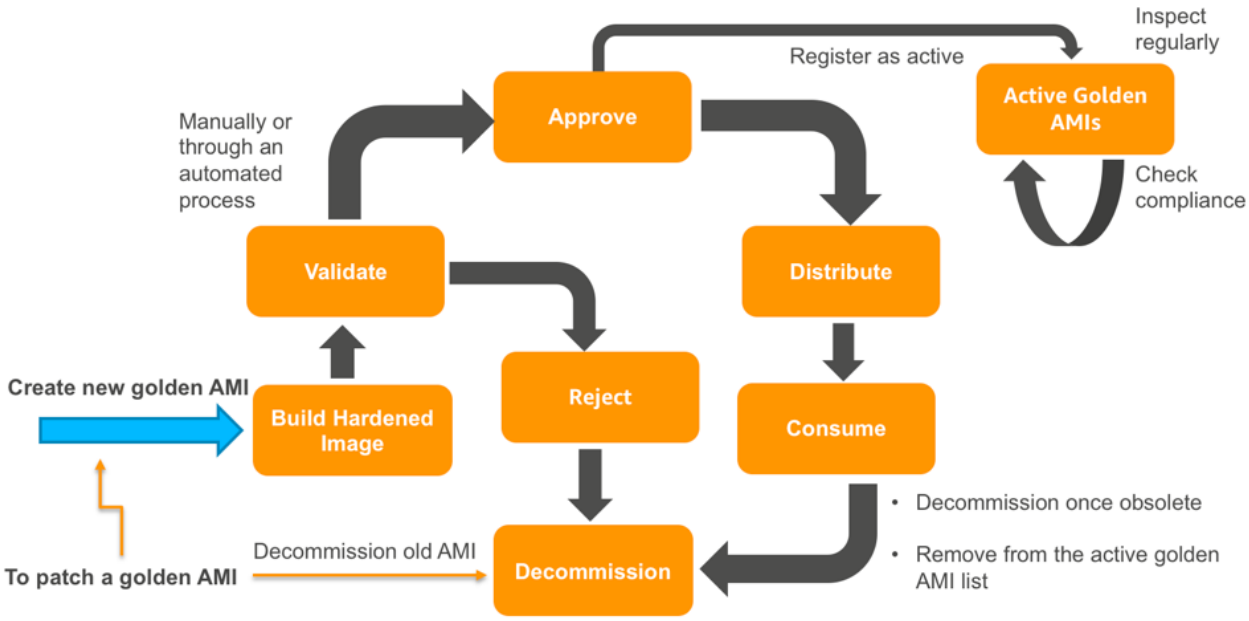
Explanation

Correct options:

Create a Golden AMI with your application

A Golden AMI is an AMI that you standardize through configuration, consistent security patching, and hardening. It also contains agents you approve for logging, security, performance monitoring, etc. For the given use-case, you can have the complex runtime and OS dependencies already setup via the golden AMI.

Golden AMI Pipeline:



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

Create a new beanstalk environment for each application and apply blue/green deployment patterns

Elastic Beanstalk provides several deployment policies and settings.

Choosing a deployment policy

Choosing the right deployment policy for your application is a tradeoff of a few considerations, and depends on your particular needs. The Deployment policies and settings page has more information about each policy, and a detailed description of the workings of some of them.

The following list provides summary information about the different deployment policies and adds related considerations.

- **All at once** – The quickest deployment method. Suitable if you can accept a short loss of service, and if quick deployments are important to you. With this method, Elastic Beanstalk deploys the new application version to each instance. Then, the web proxy or application server might need to restart. As a result, your application might be unavailable to users (or have low availability) for a short time.
- **Rolling** – Avoids downtime and minimizes reduced availability, at a cost of a longer deployment time. Suitable if you can't accept any period of completely lost service. With this method, your application is deployed to your environment one batch of instances at a time. Most bandwidth is retained throughout the deployment.
- **Rolling with additional batch** – Avoids any reduced availability, at a cost of an even longer deployment time compared to the *Rolling* method. Suitable if you must maintain the same bandwidth throughout the deployment. With this method, Elastic Beanstalk launches an extra batch of instances, then performs a rolling deployment. Launching the extra batch takes time, and ensures that the same bandwidth is retained throughout the deployment.
- **Immutable** – A slower deployment method, that ensures your new application version is always deployed to new instances, instead of updating existing instances. It also has the additional advantage of a quick and safe rollback in case the deployment fails. With this method, Elastic Beanstalk performs an immutable update to deploy your application. In an immutable update, a second Auto Scaling group is launched in your environment and the new version serves traffic alongside the old version until the new instances pass health checks.
- **Traffic splitting** – A canary testing deployment method. Suitable if you want to test the health of your new application version using a portion of incoming traffic, while keeping the rest of the traffic served by the old application version.

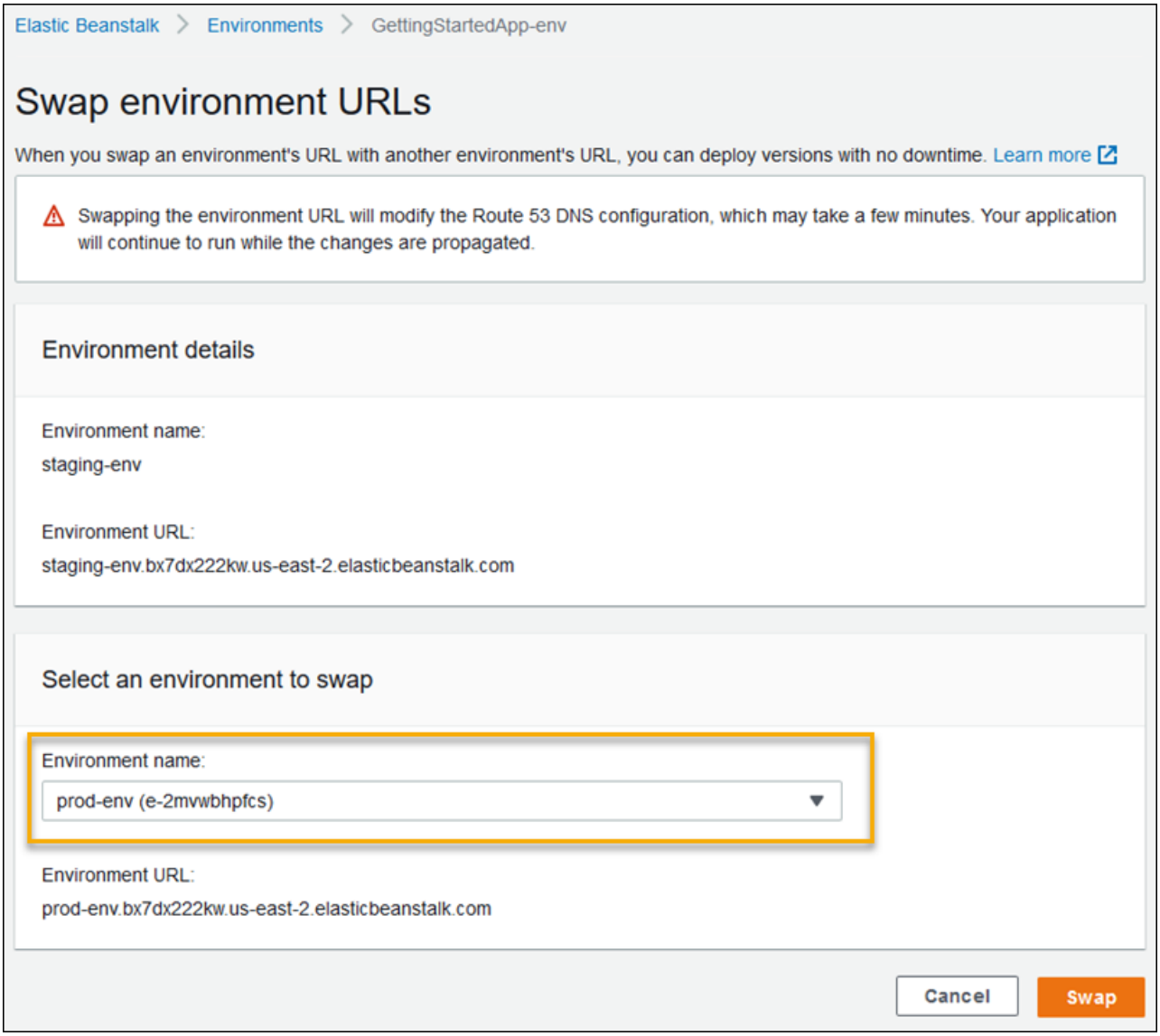
The following table compares deployment method properties.

Deployment methods

Method	Impact of failed deployment	Deploy time	Zero downtime	No DNS change	Rollback process	Code deployed to
All at once	Downtime	◷	⊗ No	⊘ Yes	Manual redeploy	Existing instances
Rolling	Single batch out of service; any successful batches before failure running new application version	◷ ◷ †	⊘ Yes	⊘ Yes	Manual redeploy	Existing instances
Rolling with an additional batch	Minimal if first batch fails; otherwise, similar to **Rolling**	◷ ◷ ◷ †	⊘ Yes	⊘ Yes	Manual redeploy	New and existing instances
Immutable	Minimal	◷ ◷ ◷ ◷	⊘ Yes	⊘ Yes	Terminate new instances	New instances
Traffic splitting	Percentage of client traffic routed to new version temporarily impacted	◷ ◷ ◷ ◷ ††	⊘ Yes	⊘ Yes	Reroute traffic and terminate new instances	New instances
Blue/green	Minimal	◷ ◷ ◷ ◷	⊘ Yes	⊗ No	Swap URL	New instances

via - https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html

Since AWS Elastic Beanstalk performs an in-place update when you update your application versions, your application can become unavailable to users for a short period of time. You can avoid this downtime by performing a blue/green deployment, where you deploy the new version to a separate environment, and then swap CNAMEs of the two environments to redirect traffic to the new version instantly.



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

A blue/green deployment is also required when you want to update an environment to an incompatible platform version.

Incorrect options:

Use rolling with additional batch - With the rolling update, your application is deployed to your environment one batch of instances at a time. Most bandwidth is retained throughout the deployment. This also avoids downtime and minimizes reduced availability, at a cost of a longer deployment time. Since the use-case mandates a short deployment time, this option is ruled out.

Upgrade the EC2 instance type - An upgraded instance type may only marginally improve the deployment time.

Use all at once deployment pattern - With all at once deployment, Elastic Beanstalk deploys the new application version to each instance. Then, the web proxy or application server might need to restart. As a result, your application might be unavailable to users (or have low availability) for a short time. Since the use-case mandates high availability, this option is ruled out.

References:

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

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

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html

Question 37: **Correct**

When you launched as a short term rental company, you had 5 employees all working on the same AWS cloud account. These employees deployed their applications for various purposes, including billing, operations, finance, etc. Each of these employees has been operating in their own VPC. Now that you have grown to over 200 employees, some employees belonging to different teams have created VPC peering connections and interfered with each other's work. You would like to properly separate the environment your employees work in based on the department they belong to.

What's the best way of achieving that?

- ○ **AWS IAM Groups with restrictive policies**
- ○ **AWS Organizations with OU** **(Correct)**
- ○ **AWS GuardDuty**
- ○ **AWS IAM Roles with restrictive policies**

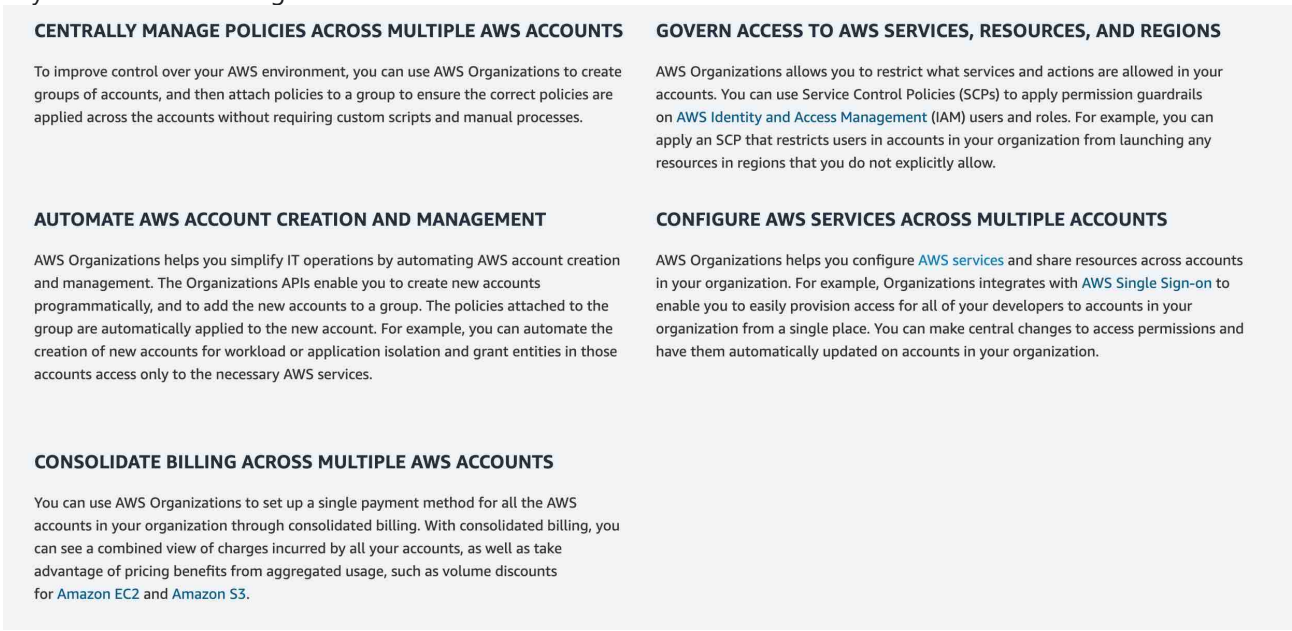
Explanation

Correct option:

AWS Organizations with OU

AWS Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts. Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance. You can also simplify billing by setting up a single payment method for all of your AWS accounts. AWS Organizations is available to all AWS customers at no additional charge.

Key Features of AWS Organizations:



CENTRALLY MANAGE POLICIES ACROSS MULTIPLE AWS ACCOUNTS

To improve control over your AWS environment, you can use AWS Organizations to create groups of accounts, and then attach policies to a group to ensure the correct policies are applied across the accounts without requiring custom scripts and manual processes.

GOVERN ACCESS TO AWS SERVICES, RESOURCES, AND REGIONS

AWS Organizations allows you to restrict what services and actions are allowed in your accounts. You can use Service Control Policies (SCPs) to apply permission guardrails on AWS Identity and Access Management (IAM) users and roles. For example, you can apply an SCP that restricts users in accounts in your organization from launching any resources in regions that you do not explicitly allow.

AUTOMATE AWS ACCOUNT CREATION AND MANAGEMENT

AWS Organizations helps you simplify IT operations by automating AWS account creation and management. The Organizations APIs enable you to create new accounts programmatically, and to add the new accounts to a group. The policies attached to the group are automatically applied to the new account. For example, you can automate the creation of new accounts for workload or application isolation and grant entities in those accounts access only to the necessary AWS services.

CONFIGURE AWS SERVICES ACROSS MULTIPLE ACCOUNTS

AWS Organizations helps you configure AWS services and share resources across accounts in your organization. For example, Organizations integrates with AWS Single Sign-on to enable you to easily provision access for all of your developers to accounts in your organization from a single place. You can make central changes to access permissions and have them automatically updated on accounts in your organization.

CONSOLIDATE BILLING ACROSS MULTIPLE AWS ACCOUNTS

You can use AWS Organizations to set up a single payment method for all the AWS accounts in your organization through consolidated billing. With consolidated billing, you can see a combined view of charges incurred by all your accounts, as well as take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

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

An Organization Unit(OU) is a container for accounts within a root. An OU also can contain other OUs, enabling you to create a hierarchy that resembles an upside-down tree, with a root at the top and branches of OUs that reach down, ending in accounts that are the leaves of the tree. When you attach a policy to one of the nodes in the hierarchy, it flows down and affects all the branches (OUs) and leaves (accounts) beneath it. An OU can have exactly one parent, and currently, each account can be a member of exactly one OU.

via - https://docs.aws.amazon.com/organizations/latest/userguide/orgs_getting-started_concepts.html

You can use AWS Organizations with OU to segregate the environment your employees work in based on the department they belong to.

Incorrect options:

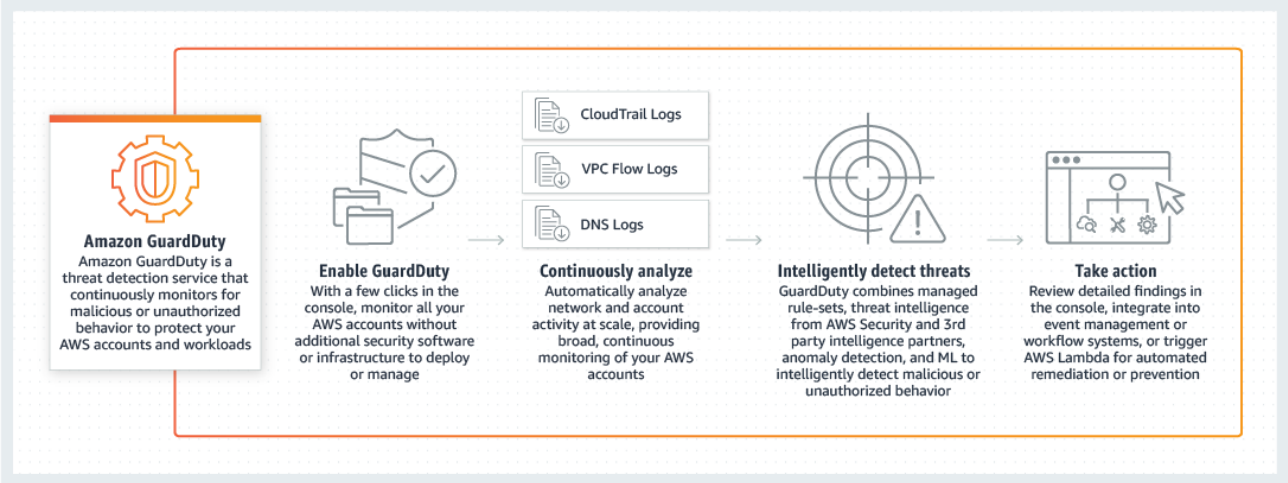
AWS IAM Groups with restrictive policies

AWS IAM Roles with restrictive policies

IAM groups or IAM roles (even with restrictive policies) cannot be used to create a segregated environment on AWS, so both these options are incorrect.

AWS GuardDuty - GuardDuty is a threat detection service that monitors malicious activity and unauthorized behavior to protect your AWS account. GuardDuty analyzes billions of events across your AWS accounts from AWS CloudTrail (AWS user and API activity in your accounts), Amazon VPC Flow Logs (network traffic data), and DNS Logs (name query patterns). GuardDuty cannot be used to segregate the environment your employees work in based on the department they belong to.

How GuardDuty Works:



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

References:

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_getting-started_concepts.html

https://aws.amazon.com/organizations/

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

Question 38: **Correct**

A company manages multiple applications on a fleet of Amazon EC2 instances. The company is looking at automating the process of patch management for all the instances that includes OS updates, application updates and security updates.

Which service/tool is the right fit for this requirement?

○ **Patch Manager, a capability of AWS Systems Manager, can be used to automate patch** **(Correct)**

~~management and OS updates for all the instances at one go~~

- ~~OS updates and patch management are responsibilities of AWS as per AWS Shared Responsibility model and does not need customer inputs~~
- ~~Patch Fleet, a capability of AWS Systems Manager, can be used to automate patch management and OS updates for all the instances at one go~~
- ~~Fleet Manager, a capability of AWS Systems Manager, can be used to automate patch management and OS updates for all the instances at one go~~

Explanation

Correct option:

Patch Manager, a capability of AWS Systems Manager, can be used to automate patch management and OS updates for all the instances at one go - Patch Manager, a capability of AWS Systems Manager, automates the process of patching managed instances with both security-related and other types of updates. You can use Patch Manager to apply patches for both operating systems and applications. (On Windows Server, application support is limited to updates for applications released by Microsoft.)

You can use Patch Manager to install Service Packs on Windows instances and perform minor version upgrades on Linux instances. You can patch fleets of Amazon Elastic Compute Cloud (Amazon EC2) instances or your on-premises servers and virtual machines (VMs) by operating system type. This includes supported versions of Amazon Linux, Amazon Linux 2, CentOS, Debian Server, macOS, Oracle Linux, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Ubuntu Server, and Windows Server. You can scan instances to see only a report of missing patches, or you can scan and automatically install all missing patches.

Incorrect options:

OS updates and patch management are responsibilities of AWS as per AWS Shared Responsibility model and does not need customer inputs - AWS Shared Responsibility Model states that - Customers that deploy an Amazon EC2 instance are responsible for the management of the guest operating system (including updates and security patches), any application software or utilities installed by the customer on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance. Therefore, this option is incorrect.

Fleet Manager, a capability of AWS Systems Manager, can be used to automate patch management and OS updates for all the instances at one go - Fleet Manager, a capability of AWS Systems Manager, is a unified user interface (UI) experience that helps you remotely manage your server fleet running on AWS, or on-premises. With Fleet Manager, you can view the health and performance status of your entire server fleet from one console. You can also gather data from individual instances to perform common troubleshooting and management tasks from

the console. This includes viewing folder and file contents, Windows registry management, operating system user management, and more.

Patch Fleet, a capability of AWS Systems Manager, can be used to automate patch management and OS updates for all the instances at one go - There is no such thing as Patch Fleet. This option has been added as a distractor.

References:

https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html

https://aws.amazon.com/compliance/shared-responsibility-model/

Question 39: **Correct**

An AWS Lambda function written in Python shuts down all instances at night for cost savings purposes. Some of these instances should not be shut down, as the underlying applications transition to an unstable state afterward.

How could you efficiently prevent the shut down of the critical instances?

- ○ **Change the shutdown behavior of the EC2 instances and enable termination protection as well**
- ○ **Store all the instance ids you should not shut down in SSM Parameter Store**
- ○ **Use an environment variable for your AWS Lambda with a list of instances not to shut down**
- ○ **Tag your EC2 instances and make the AWS Lambda script skip the shutdown if the tag is found** **(Correct)**

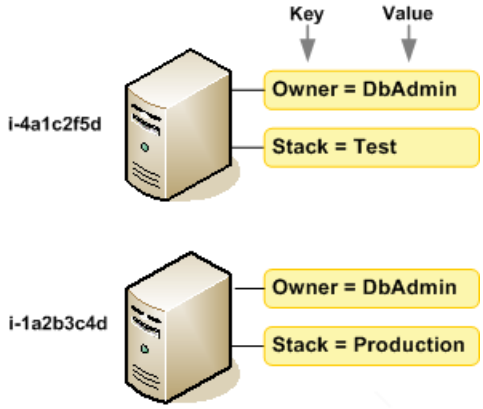
Explanation

Correct option:

Tag your EC2 instances and make the AWS Lambda script skip the shutdown if the tag is found

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. For example, you could define a set of tags for your account's Amazon EC2 instances that help you track each instance's owner and stack level.

AWS Resource Tags: 

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

Tags allow for the cleanest solution here, as you can create instances in the future and tag them accordingly at the time of creation, without the need to modify your AWS Lambda function.

Incorrect options:

Store all the instance ids you should not shut down in SSM Parameter Store - AWS Systems Manager Parameter Store (aka SSM Parameter Store) provides secure, hierarchical storage for configuration data management and secrets management. You can store data such as passwords, database strings, EC2 instance IDs, Amazon Machine Image (AMI) IDs, and license codes as parameter values. You can store values as plain text or encrypted data. You can reference Systems Manager parameters in your scripts, commands, SSM documents, and configuration and automation workflows by using the unique name that you specified when you created the parameter.

You would need to modify the parameter store values every time you need to add/delete/modify the instances, so this option is not the right fit for the given use-case.

Use an environment variable for your AWS Lambda with a list of instances not to shut down - An environment variable is a pair of strings that are stored in a function's version-specific configuration. The Lambda runtime makes environment variables available to your code and sets additional environment variables that contain information about the function and invocation request.

You would need to modify the environment variables every time you need to add/delete/modify the instances, so this option is not the right fit for the given use-case.

Change the shutdown behavior of the EC2 instances and enable termination protection as well - By default, you can terminate your instance using the Amazon EC2 console, command line interface, or API. To prevent your instance from being accidentally terminated using Amazon EC2, you can enable termination protection for the instance.

You would need to modify the termination behavior every time you need to add/delete/modify an instance running the application, so this option is not the right fit for the given use-case.

References:

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html#Using_ChangingDisableAPITermination

Question 40: **Correct**

You plan on creating a subnet and want it to have at least capacity for 28 EC2 instances.

What's the minimum size you need to have for your subnet?

◯ /26 **(Correct)**

/27

/28

/25

Explanation

Correct option:

/26

You should note that the first four IP addresses and the last IP address in each subnet CIDR block aren't available for your use. These 5 IP addresses can't be assigned to an instance.

The formula for a "/x" subnet = (2 * power(32-x)) - 5

Therefore, for a "/26" subnet, you have:

= (2 * power(32-26)) - 5

= (2 * power(6)) - 5

= 64 - 5

= 59

So you have 59 IP addresses, which meets the given requirement of having a capacity for at least 28 EC2 instances.

Incorrect options:

/28

/27

/25

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

References:

https://aws.amazon.com/premiumsupport/knowledge-center/change-subnet-mask/

https://s3.amazonaws.com/tr-learncanvas/docs/IP_Filtering_in_Canvas.pdf

Question 41: **Correct**

As a service provider, you generate a daily report that you need to share with your dynamically changing list of over 10,000 customers. These reports sit in S3, and you would like to automate sharing the reports with them so they can have on-demand access upon their identity being proven.

You plan to use Cognito, API Gateway and AWS Lambda to address this use-case. On the S3 side, what should you do?

- ○ **Make the S3 bucket public and password protect each S3 file. Share the password with each customer**
- ○ **Create a bucket policy so that the S3 files are only accessible from CloudFront and force SSL mutual authentication there**
- ○ **Provide each of your customers an AWS user and tell them to use the CLI**
- ○ **Generate pre-signed URLs for your reports** **(Correct)**

Explanation

Correct option:

Generate pre-signed URLs for your reports

A presigned URL gives you access to the object identified in the URL, provided that the creator of the presigned URL has permissions to access that object.

All objects by default are private. Only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a presigned URL, using their own security credentials, to grant time-limited permission to download the objects.

When you create a presigned URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The presigned URLs are valid only for the specified duration.

Anyone who receives the presigned URL can then access the object. For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a presigned URL.

Incorrect options:

Provide each of your customers an AWS user and tell them to use the CLI - This is not practicable considering that there are 10,000 customers.

Create a bucket policy so that the S3 files are only accessible from CloudFront and force SSL mutual authentication there - Mutual Transport Layer Security (TLS) authentication is supported for Amazon API Gateway and not for CloudFront. This is a new method for client-to-server authentication that can be used with API Gateway's existing authorization options.

Make the S3 bucket public and password protect each S3 file. Share the password with each customer - This is a distractor as there is no way to password protect files on S3.

Reference:

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

Question 42: **Correct**

A healthcare company has machines both on their own data center for HIPAA compliance reasons, as well as on the AWS cloud to perform their big data analysis. All the instances must be managed using the same Puppet modules, as per the CTO decision.

Which AWS service helps you in achieving that?

- ◯ **OpsWorks** **(Correct)**
- ◯ **Artifact**
- ◯ **GuardDuty**
- ◯ **Ansible**

Explanation

Correct option:

OpsWorks

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed and managed across your Amazon EC2 instances or on-premises compute environments.

AWS OpsWorks for Puppet Enterprise provides a managed Puppet Enterprise server and suite of automation tools that give you workflow automation for orchestration, automated provisioning, and visualization for traceability. The Puppet Enterprise server gives you full stack automation by handling operational tasks such as software and operating system configurations, package installations, database setups, and more. The Puppet Master centrally stores your configuration tasks and provides them to each node in your compute environment at any scale, from a few nodes to thousands of nodes.

Incorrect options:

Artifact - AWS Artifact is a self-service audit artifact retrieval portal that provides our customers with on-demand access to AWS' compliance documentation and AWS agreements. You cannot use Artifact to deploy the patches on the instance. You cannot use Artifact to manage the instances using Puppet modules.

GuardDuty - Amazon GuardDuty offers threat detection that enables you to continuously monitor and protect your AWS accounts, workloads, and data stored in Amazon S3. GuardDuty analyzes continuous streams of meta-data generated from your account and network activity found in AWS CloudTrail Events, Amazon VPC Flow Logs, and DNS Logs. It also uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning to identify threats more accurately. You cannot use GuardDuty to manage the instances using Puppet modules.

Ansible - Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. You cannot use Ansible to manage the instances using Puppet

modules.

Reference:

https://aws.amazon.com/opsworks/

Question 43: **Correct**

You have developed a script that checks if all the instances that were launched in your AWS region are using an AMI ID that is authorized by your financial company standards. After creating and testing this script in your region, eu-west-1, you share it with your colleagues in New York and ask them to run the script. Upon running it, they come back to you and say it's not working, as all the instances are declared non-compliant. Auditors manually checked the instances and they are indeed compliant.

What did you do wrong?

- ○ **The API call limit has been reached and the script did not handle that error case**
- ○ **AMI IDs are region-specific and a different list of compliant AMI ID should be provided based on the region of where the script is executed** **(Correct)**
- ○ **Your colleagues did not run the script properly. You write detailed documentation on what they did wrong**
- ○ **The script is missing IAM permissions. Edit the script to include the IAM policy from within and run it again**

Explanation

Correct option: **AMI IDs are region-specific and a different list of compliant AMI ID should be provided based on the region of where the script is executed**

An Amazon Machine Image (AMI) provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

As the AMI is tied to a specific AWS Region, you need to copy the AMI to other AWS Regions if required. You can copy an Amazon Machine Image (AMI) within or across AWS Regions using the AWS Management Console, the AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support the CopyImage action. You can copy both Amazon EBS-backed AMIs and instance-store-backed AMIs. You can copy AMIs with encrypted snapshots and also change encryption status during the copy process.

Copying a source AMI results in an identical but distinct target AMI with its own unique identifier. In the case of an Amazon EBS-backed AMI, each of its backing snapshots is, by default, copied to an identical but distinct target snapshot. (The sole exceptions are when you choose to encrypt or re-encrypt the snapshot.) You can change or deregister the source AMI with no effect on the target AMI.

For the given use-case, you need to provide the unique identifiers for the AMIs created in other AWS regions so that those can be used in the validation script.

Incorrect options:

The script is missing IAM permissions. Edit the script to include the IAM policy from within and run it again

The API call limit has been reached and the script did not handle that error case

Your colleagues did not run the script properly. You write detailed documentation on what they did wrong

These three options contradict the explanation given above, therefore these are incorrect.

Reference:

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

Question 44: **Correct**

You want a small website on EC2 instances under an ASG that has a target size varying between 2 and 10 instances. Your ASG has a policy to scale out when your target CPU Utilization is above 75%. It has been over 3 hours that the CPU Utilization of your ASG is 90% and still, no scaling out actions have taken place.

What are the most likely reasons for this? (Select two)

Your ASG AZRebalance process has been suspended

Your ASG Launch process has been suspended **(Correct)**

The warmup period of the EC2 instances has not elapsed yet

Your ASG is at maximum capacity already (Correct)

AWS does not have the capacity for more of the requested EC2 instance types

Explanation

Correct options:

A scaling policy instructs Amazon EC2 Auto Scaling to track a specific CloudWatch metric, and it defines what action to take when the associated CloudWatch alarm is in ALARM. The metrics that are used to trigger an alarm are an aggregation of metrics coming from all of the instances in the Auto Scaling group. (For example, let's say you have an Auto Scaling group with two instances where one instance is at 60 percent CPU and the other is at 40 percent CPU. On average, they are at 50 percent CPU.) When the policy is in effect, Amazon EC2 Auto Scaling adjusts the group's desired capacity up or down when the alarm is triggered.

Scaling policy types

Amazon EC2 Auto Scaling supports the following types of scaling policies:

- **Target tracking scaling**—Increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home—you select a temperature and the thermostat does the rest.
- **Step scaling**—Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as *step adjustments*, that vary based on the size of the alarm breach.
- **Simple scaling**—Increase or decrease the current capacity of the group based on a single scaling adjustment.

If you are scaling based on a utilization metric that increases or decreases proportionally to the number of instances in an Auto Scaling group, we recommend that you use target tracking scaling policies. Otherwise, we recommend that you use step scaling policies.

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html

Your ASG is at maximum capacity already

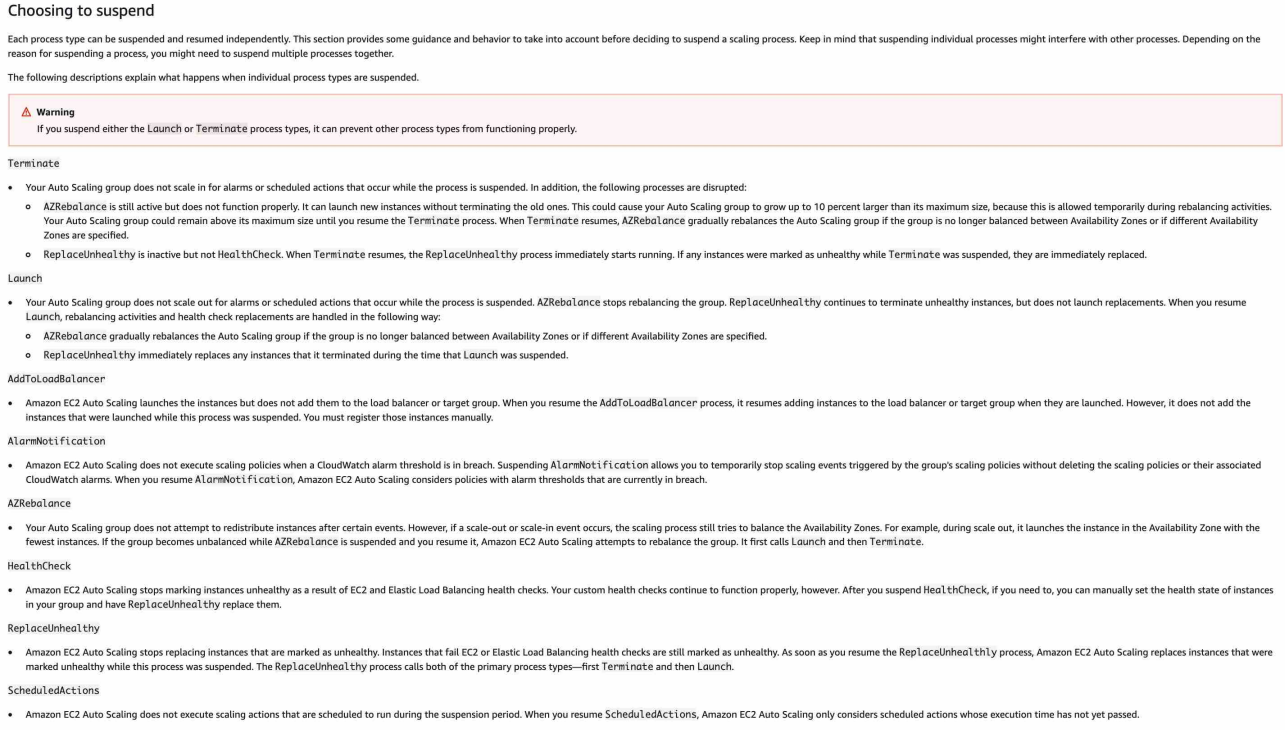
You can configure the size of your Auto Scaling group by setting the minimum, maximum, and desired capacity. The minimum and maximum capacity are required to create an Auto Scaling group, while the desired capacity is optional. If you do not define your desired capacity upfront, it defaults to your minimum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

If your ASG is already at the maximum capacity, then it will not lead to a scale out action.

Your ASG Launch process has been suspended

For Amazon EC2 Auto Scaling, there are two primary process types: Launch and Terminate. The Launch process adds a new Amazon EC2 instance to an Auto Scaling group, increasing its capacity. The Terminate process removes an Amazon EC2 instance from the group, decreasing its capacity.

If the Launch process is suspended, then your Auto Scaling group does not scale out for alarms or any scheduled actions that occur.



Choosing to suspend

Each process type can be suspended and resumed independently. This section provides some guidance and behavior to take into account before deciding to suspend a scaling process. Keep in mind that suspending individual processes might interfere with other processes. Depending on the reason for suspending a process, you might need to suspend multiple processes together.

The following descriptions explain what happens when individual process types are suspended.

⚠ **Warning**
If you suspend either the `Launch` or `Terminate` process types, it can prevent other process types from functioning properly.

`Terminate`

- Your Auto Scaling group does not scale in for alarms or scheduled actions that occur while the process is suspended. In addition, the following processes are disrupted:
  - `AZRebalance` is still active but does not function properly. It can launch new instances without terminating the old ones. This could cause your Auto Scaling group to grow up to 10 percent larger than its maximum size, because this is allowed temporarily during rebalancing activities. Your Auto Scaling group could remain above its maximum size until you resume the `Terminate` process. When `Terminate` resumes, `AZRebalance` gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
  - `ReplaceUnhealthy` is inactive but not `HealthCheck`. When `Terminate` resumes, the `ReplaceUnhealthy` process immediately starts running. If any instances were marked as unhealthy while `Terminate` was suspended, they are immediately replaced.

`Launch`

- Your Auto Scaling group does not scale out for alarms or scheduled actions that occur while the process is suspended. `AZRebalance` stops rebalancing the group. `ReplaceUnhealthy` continues to terminate unhealthy instances, but does not launch replacements. When you resume `Launch`, rebalancing activities and health check replacements are handled in the following way:
  - `AZRebalance` gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
  - `ReplaceUnhealthy` immediately replaces any instances that it terminated during the time that `Launch` was suspended.

`AddToLoadBalancer`

- Amazon EC2 Auto Scaling launches the instances but does not add them to the load balancer or target group. When you resume the `AddToLoadBalancer` process, it resumes adding instances to the load balancer or target group when they are launched. However, it does not add the instances that were launched while this process was suspended. You must register those instances manually.

`AlarmNotification`

- Amazon EC2 Auto Scaling does not execute scaling policies when a CloudWatch alarm threshold is in breach. Suspending `AlarmNotification` allows you to temporarily stop scaling events triggered by the group's scaling policies without deleting the scaling policies or their associated CloudWatch alarms. When you resume `AlarmNotification`, Amazon EC2 Auto Scaling considers policies with alarm thresholds that are currently in breach.

`AZRebalance`

- Your Auto Scaling group does not attempt to redistribute instances after certain events. However, if a scale-out or scale-in event occurs, the scaling process still tries to balance the Availability Zones. For example, during scale out, it launches the instance in the Availability Zone with the fewest instances. If the group becomes unbalanced while `AZRebalance` is suspended and you resume it, Amazon EC2 Auto Scaling attempts to rebalance the group. It first calls `Launch` and then `Terminate`.

`HealthCheck`

- Amazon EC2 Auto Scaling stops marking instances unhealthy as a result of EC2 and Elastic Load Balancing health checks. Your custom health checks continue to function properly, however. After you suspend `HealthCheck`, if you need to, you can manually set the health state of instances in your group and have `ReplaceUnhealthy` replace them.

`ReplaceUnhealthy`

- Amazon EC2 Auto Scaling stops replacing instances that are marked as unhealthy. Instances that fail EC2 or Elastic Load Balancing health checks are still marked as unhealthy. As soon as you resume the `ReplaceUnhealthly` process, Amazon EC2 Auto Scaling replaces instances that were marked unhealthy while this process was suspended. The `ReplaceUnhealthy` process calls both of the primary process types—first `Terminate` and then `Launch`.

`ScheduledActions`

- Amazon EC2 Auto Scaling does not execute scaling actions that are scheduled to run during the suspension period. When you resume `ScheduledActions`, Amazon EC2 Auto Scaling only considers scheduled actions whose execution time has not yet passed.

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html

Incorrect options:

Your ASG AZRebalance process has been suspended - For the AZRebalance process type, your Auto Scaling group does not attempt to redistribute instances after certain events. However, if a scale-out or scale-in event occurs, the scaling process still tries to balance the Availability Zones. Suspending the AZRebalance process will not stop the scale out from happening.

AWS does not have the capacity for more of the requested EC2 instance types - If this is the case, you will get an error message that says "Your requested instance type (<instance type>) is not supported in your requested Availability Zone (<instance Availability Zone>). Please retry your request by not specifying an Availability Zone or choosing <list of Availability Zones that supports the instance type>. Launching EC2 instance failed.".

The warmup period of the EC2 instances has not elapsed yet - The warmup period specifies the number of seconds that it takes for a newly launched instance to warm up. Until its specified warm-up time has expired, an instance is not counted toward the aggregated metrics of the Auto Scaling group. If the warmup period was set to a high value, then new instances should still have been launched as the existing instance under warmup would not have contributed to the aggregated metrics yet. So this option is incorrect.

References:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/asg-capacity-limits.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/ts-as-instancelaunchfailure.html

Question 45: **Correct**

How can you enforce encryption on all the files uploaded into your example S3 bucket?

- Use an encrypted CloudFront distribution in front of your S3 bucket

- Using the "Default Encryption" setting in AWS S3 (Correct)

- Use the following S3 bucket policy:

```
{
    "Statement":[
        {
            "Action": "s3:*",
            "Effect":"Deny",
            "Principal": "*",
            "Resource":"arn:aws:s3:::bucketname/*",
            "Condition":{
                "Bool":
                { "aws:SecureTransport": false }
            }
        }
    ]
}
```

- Use the following S3 bucket policy:

```
{
    "Statement":[
        {
            "Action": "s3:*",
            "Effect":"Deny",
            "Principal": "*",
            "Resource":"arn:aws:s3:::bucketname/*",
            "Condition":{
                "Bool":
                { "aws:SecureTransport": true }
            }
        }
    ]
}
```

Explanation

Correct option:

Using the "Default Encryption" setting in AWS S3

Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all new objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or customer master keys (CMKs) stored in AWS Key Management Service (AWS KMS).

When you use server-side encryption, Amazon S3 encrypts an object before saving it to disk and decrypts it when you download the objects.

Incorrect options:

Use the following S3 bucket policy:

```
{
    "Statement":[
        {
            "Action": "s3:*",
            "Effect":"Deny",
            "Principal": "*",
            "Resource":"arn:aws:s3:::bucketname/*",
            "Condition":{
                "Bool":
                { "aws:SecureTransport": false }
            }
        }
    ]
}
```

The above bucket policy only denies access to HTTP requests for any action on the S3 bucket `bucketname`. It cannot help enforce SSE-S3 encryption on S3. So it's not the right fit for the given use-case.

Use the following S3 bucket policy:

```
{
    "Statement":[
        {
            "Action": "s3:*",
            "Effect":"Deny",
            "Principal": "*",
            "Resource":"arn:aws:s3:::bucketname/*",
            "Condition":{
                "Bool":
                { "aws:SecureTransport": true }
            }
        }
    ]
}
```

The above bucket policy only denies access to HTTPS requests for any action on the S3 bucket `bucketname`. It cannot help enforce SSE-S3 encryption on S3. So it's not the right fit for the given use-case.

Use an encrypted CloudFront distribution in front of your S3 bucket - This option is a distractor as you cannot enforce SSE-S3 encryption on S3 by using in-transit or at-rest encryption for CloudFront.

References:

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

https://aws.amazon.com/blogs/security/how-to-prevent-uploads-of-unencrypted-objects-to-amazon-s3/

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/data-protection-summary.html

Question 46: **Correct**

You have deployed a public and a private subnet. As such, you have also deployed an Internet Gateway, a NAT Gateway, an Egress Only Internet Gateway, and a Virtual Private Gateway. You would like your private subnet instances to get IPv4 access to the internet.

How should you edit your route tables?

- ○ **Add a route with a target of 0.0.0.0/0 to the NAT Gateway** **(Correct)**
- ○ **Add a route with a target of 0.0.0.0/0 to the Egress Only Internet Gateway**
- ○ **Add a route with a target of 10.0.0.0/12 to the Virtual Private Gateway**
- ○ **Add a route with a target of 0.0.0.0/0 to the Internet Gateway**

Explanation

Correct option:

Add a route with a target of 0.0.0.0/0 to the NAT Gateway

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

1. A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.
2. You can associate exactly one Elastic IP address with a NAT gateway.
3. A NAT gateway supports the following protocols: TCP, UDP, and ICMP.
4. You cannot associate a security group with a NAT gateway.

5. You can use a network ACL to control the traffic to and from the subnet in which the NAT gateway is located.
6. A NAT gateway can support up to 55,000 simultaneous connections to each unique destination.

Therefore you must use a NAT Gateway in your public subnet in order to provide internet access to your instances in your private subnets. You also need to set up the appropriate entries in the route table of the private subnets, so for the given use-case, you need to add a route with a target of 0.0.0.0/0 to the NAT Gateway. You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Incorrect options:

Add a route with a target of 0.0.0.0/0 to the Internet Gateway - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. Internet Gateways must be deployed in a public subnet and the corresponding entry should be added in the route table of the public subnet. You should set up the correct entry in the route table of the private subnet, so you need to add a route with a target of 0.0.0.0/0 to the NAT Gateway and NOT to the Internet Gateway.

Add a route with a target of 0.0.0.0/0 to the Egress Only Internet Gateway - An egress-only internet gateway is for use with IPv6 traffic only, so this option is incorrect.

Add a route with a target of 10.0.0.0/12 to the Virtual Private Gateway - A Virtual Private Gateway is the Amazon VPC side of a VPN connection, so it's not relevant to the given scenario.

Reference:

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

Question 47: **Correct**

Your accounting application on an EC2 instance has the tendency to sometimes go into a panic and then the CPU Utilization of your EC2 instance runs at 100% for a long duration. When this happens, someone has to manually intervene and restart your application for it to work properly again.

How can you automate this in the most efficient way?

- ○ **Put your instance in an ASG and behind an ELB and enable ELB health check, so that the instance gets terminated upon problems and a new one gets created**
- ○ **Create a CloudWatch Event when CPU Utilization reaches 100% and trigger an EC2 reboot action**
- ○ **Invoke an AWS Lambda function via a cron job that checks for the metric every minute and restarts the instance if a problem is found**
- ○ **Create a CloudWatch Alarm when CPU Utilization reaches 100% for 3 periods of 5 minutes and** **(Correct)**

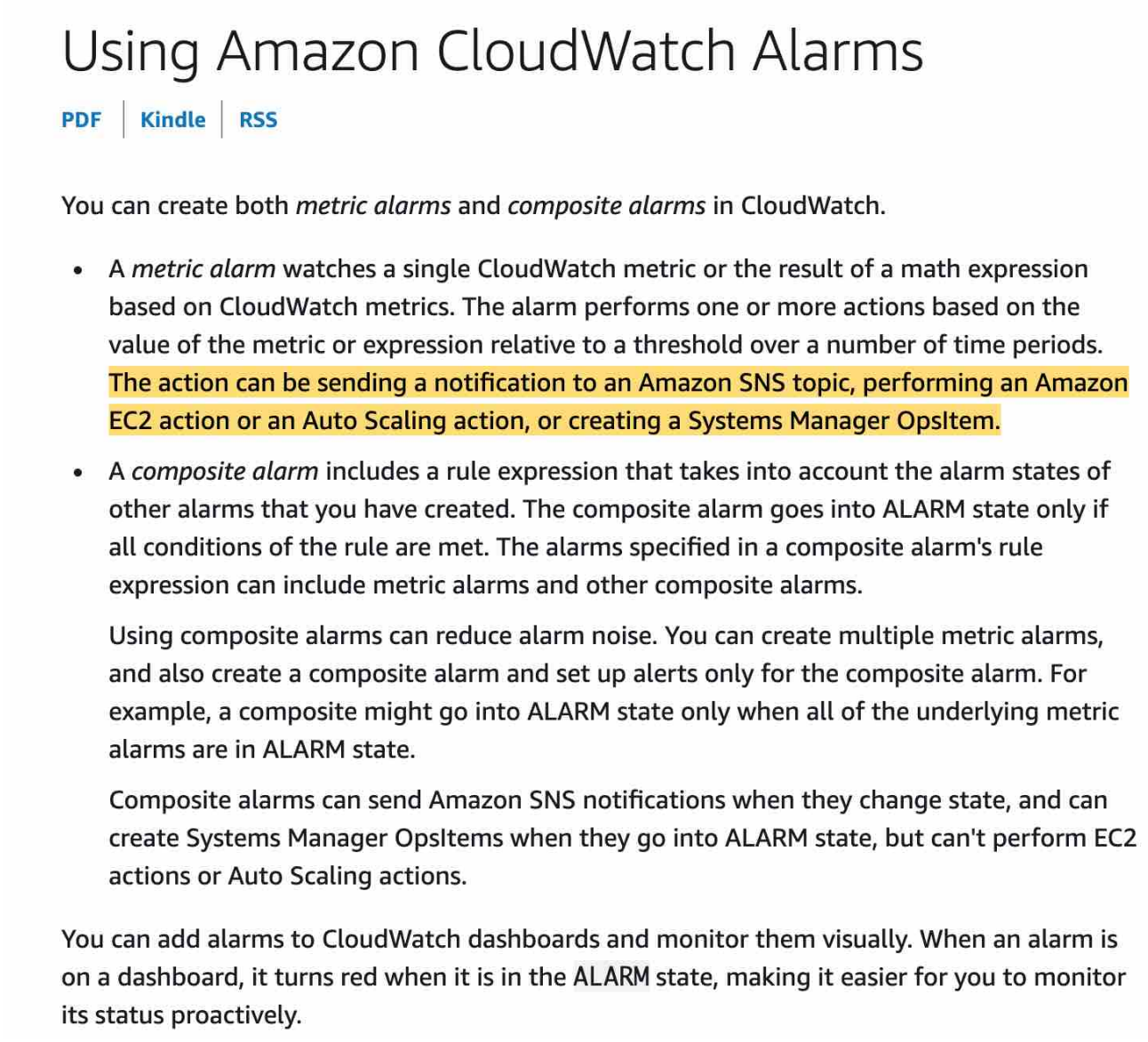
trigger an EC2 reboot action

Explanation

Correct option:

Create a CloudWatch Alarm when CPU Utilization reaches 100% for 3 periods of 5 minutes and trigger an EC2 reboot action

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your EC2 instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.



Using Amazon CloudWatch Alarms

PDF | Kindle | RSS

You can create both *metric alarms* and *composite alarms* in CloudWatch.

- A *metric alarm* watches a single CloudWatch metric or the result of a math expression based on CloudWatch metrics. The alarm performs one or more actions based on the value of the metric or expression relative to a threshold over a number of time periods. The action can be sending a notification to an Amazon SNS topic, performing an Amazon EC2 action or an Auto Scaling action, or creating a Systems Manager OpsItem.
- A *composite alarm* includes a rule expression that takes into account the alarm states of other alarms that you have created. The composite alarm goes into ALARM state only if all conditions of the rule are met. The alarms specified in a composite alarm's rule expression can include metric alarms and other composite alarms.

  Using composite alarms can reduce alarm noise. You can create multiple metric alarms, and also create a composite alarm and set up alerts only for the composite alarm. For example, a composite might go into ALARM state only when all of the underlying metric alarms are in ALARM state.

  Composite alarms can send Amazon SNS notifications when they change state, and can create Systems Manager OpsItems when they go into ALARM state, but can't perform EC2 actions or Auto Scaling actions.

You can add alarms to CloudWatch dashboards and monitor them visually. When an alarm is on a dashboard, it turns red when it is in the ALARM state, making it easier for you to monitor its status proactively.

via - https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html

Exam Alert:

Please note that the CloudWatch Alarm action can only have the following targets and the exam may trick you into choosing other options:

A valid CloudWatch action can be sending a notification to an Amazon SNS topic, performing an Amazon EC2 action or an Auto Scaling action, or creating a Systems Manager OpsItem.

Incorrect options:

Invoke an AWS Lambda function via a cron job that checks for the metric every minute and restarts the instance if a problem is found - Invoking an AWS Lambda function via a cron job to check the metric every minute represents a wasteful and inelegant solution. Using the CloudWatch Alarms action is a better solution for the given use-case.

Create a CloudWatch Event when CPU Utilization reaches 100% and trigger an EC2 reboot action - The CloudWatch metric - CPU Utilization - for an EC2 instance can be directly consumed to set up a CloudWatch Alarm and it cannot be used to invoke a CloudWatch Event.

Put your instance in an ASG and behind an ELB and enable ELB health check, so that the instance gets terminated upon problems and a new one gets created - Using an ELB incurs additional costs and the combination of ELB with ASG adds more complexity to the solution, so it's not an efficient solution for the given use-case.

References:

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

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

Question 48: **Correct**

You sell beauty products and have spent thousands of dollars on a new marketing campaign that declares that the 22nd of February is "national beauty day". The marketing campaign is showing very early signs of success and on the 22nd of February, you expect traffic to increase by 10x on your website. Your CEO wants to make sure your entire infrastructure is ready for the big day. Your website runs on Elastic Beanstalk, which deployed an ASG and an ELB.

What should you do to ensure you can handle the traffic? (Select two)

Open a support request with AWS to pre-warm the load balancer **(Correct)**

Open a support request to increase the upper limit on the number of the EC2 instance types you're using **(Correct)**

Use a weighted policy record in Route 53

Enable Blue/Green Beanstalk Deployment

Open a support request with AWS to request a penetration testing authorization

Explanation

Correct options:

Open a support request with AWS to pre-warm the load balancer

ELB can handle the vast majority of use cases for the customers without requiring "pre-warming" (configuring the load balancer to have the appropriate level of capacity based on expected traffic). In certain scenarios, such as when flash traffic is expected, or in the case where a load test cannot be configured to gradually increase traffic, AWS recommends that you contact AWS to have your load balancer "pre-warmed". AWS will then configure the load balancer to have the appropriate level of capacity based on the traffic that you expect. AWS will need to know the start and end dates of your tests or expected flash traffic, the expected request rate per second, and the total size of the typical request/response that you will be testing.

Open a support request to increase the upper limit on the number of the EC2 instance types you're using

When you create your AWS account, AWS sets default quotas (also referred to as limits) on these resources on a per-Region basis. For example, there is a maximum number of instances that you can launch in a Region. So if you were to launch an instance in the US West (Oregon) Region, for example, the request must not cause your usage to exceed your maximum number of instances in that Region. If you get an `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance, you have reached the limit on the number of instances that you can launch in a Region.

For the given use-case, you need to create a support request to raise the upper limit on the number of allowed EC2 instances from 20 to an appropriate number that allows the ASG to scale-out to meet the usage demand.

Incorrect options:

Open a support request with AWS to request a penetration testing authorization - AWS allows its customers to carry out security assessments or penetration tests against their AWS infrastructure without prior approval for the permitted services. However, penetration testing cannot help meet the traffic demand for the given use-case.

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

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

Because AWS Elastic Beanstalk performs an in-place update when you update your application versions, your application can become unavailable to users for a short period of time. You can avoid this downtime by

performing a blue/green deployment, where you deploy the new version to a separate environment, and then swap CNAMEs of the two environments to redirect traffic to the new version instantly.

Blue/Green Beanstalk Deployment will not help meet the traffic demand for the given use-case.

Use a weighted policy record in Route 53 - Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of the software.

Weighted routing will not help meet the traffic demand for the given use-case.

References:

https://aws.amazon.com/articles/best-practices-in-evaluating-elastic-load-balancing/#pre-warming

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/troubleshooting-launch.html

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.deploy-existing-version.html

https://aws.amazon.com/security/penetration-testing/

Question 49: **Correct**

You are developing a new CloudFormation stack and writing some very complex cfn-init code. The code fails and you would like to debug why. When reading the documentation, you see all the logs are in the file `/var/cfn/cfn-init-output.log` and will give you more information as to why the instance provisioning is failing. But you realize that you can't gain access to this file as the CloudFormation stack always terminates the EC2 instance when the creation fails.

What can you do to access these logs files, while not changing the way your EC2 instance works and ensuring you can debug your instance over 24 hours?

- ◯ **Set OnFailure=DO_NOTHING** **(Correct)**
- ◯ **Increase the Wait Timeout to 2 hours**
- ◯ **Install the CloudWatch logs agent, create a new IAM role and assign it to the EC2 instance, and send the logs directly to CloudWatch Logs**
- ◯ **Enable VPC Flow Logs and intercept the cfn-init log file**

Explanation

Correct option:

Set OnFailure=DO_NOTHING

You can use the OnFailure property of the CloudFormation CreateStack call for this use-case. The OnFailure property determines what action will be taken if stack creation fails. This must be one of DO_NOTHING, ROLLBACK, or DELETE. You can specify either OnFailure or DisableRollback, but not both.

Using the OnFailure property, you can prevent the termination of the EC2 instances created by the CloudFormation stack.

Incorrect options:

Install the CloudWatch logs agent, create a new IAM role and assign it to the EC2 instance, and send the logs directly to CloudWatch Logs

Enable VPC Flow Logs and intercept the cfn-init log file

As the use-case mentions that there should be no changes done to the EC2 instance, so both these options are ruled out since these involve installing or configuring additional software.

Increase the Wait Timeout to 2 hours - The wait timeout works with cfn-signal, however, the given issue is related to cfn-init wherein some underlying code is failing. Therefore increasing wait timeout is not a valid solution for this scenario.

References:

https://docs.aws.amazon.com/AWSCloudFormation/latest/APIReference/API_CreateStack.html

https://aws.amazon.com/premiumsupport/knowledge-center/cloudformation-prevent-rollback-failure/

Question 50: **Correct**

Your RDS database sometimes can become unresponsive, failing health checks and you need your application to fail-over automatically and safely without losing any committed transactions.

Which options would you choose?

- ○ **Setup a CloudWatch alarm for DB RAM going over 90% and reboot the database then**
- ○ **Enable RDS Multi-AZ** **(Correct)**
- ○ **Create an RDS read replica in the same region and an AWS lambda function to promote that replica as the main database when the main RDS database is down**
- ○ **Create an RDS read replica in a different region and an AWS lambda function to promote that replica as the main database when the main RDS database is down**

Explanation

Correct option:

Enable RDS Multi-AZ

RDS provides high availability and failover support for DB instances using Multi-AZ deployments. In a Multi-AZ deployment, RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone.

The failover happens only in the following conditions:

The primary DB instance fails

An Availability Zone outage

The DB instance server type is changed

The operating system of the DB instance is undergoing software patching.

A manual failover of the DB instance can be initiated using Reboot with failover.

High availability (Multi-AZ) for Amazon RDS

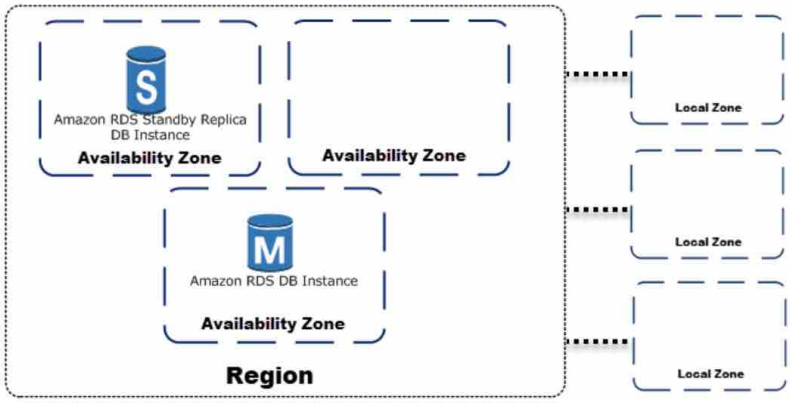
PDF | Kindle | RSS

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

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

Note

The high-availability feature is not a scaling solution for read-only scenarios; you cannot use a standby replica to serve read traffic. To service read-only traffic, you should use a read replica. For more information, see Working with read replicas.



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

Incorrect options:

Create an RDS read replica in the same region and an AWS lambda function to promote that replica as the main database when the main RDS database is down

Create an RDS read replica in a different region and an AWS lambda function to promote that replica as the main database when the main RDS database is down

RDS Read Replicas provide enhanced performance and durability for RDS database (DB) instances. They make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. Whether you create the Read Replica in the same AWS Region or a different Region from the primary DB, you cannot use it for building a High Availability solution that also transparently switches to the standby by

maintaining the same DB endpoint. Using AWS Lambda to promote the Read Replica as the main database when the primary RDS database is down, would cause the DB endpoint to change.

Therefore, both of these options are incorrect.

Exam Alert:

Please review the key differences between Read Replicas and Multi-AZ:

Read replicas, Multi-AZ deployments, and multi-region deployments

Amazon RDS read replicas complement Multi-AZ deployments. While both features maintain a second copy of your data, there are differences between the two:

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

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

Setup a CloudWatch alarm for DB RAM going over 90% and reboot the database then - This is akin to applying a band-aid. As the root cause persists, as soon as the DB instance is up and running, you will face the DB performance issues again.

References:

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

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

Question 51: **Correct**

Which of the following services allows for an in-place switch from unencrypted to encrypted without impacting existing operations?

- ○ **EBS**
- ○ **EFS**
- ○ **S3** **(Correct)**

◯ **RDS**

Explanation

Correct options:

S3

Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all new objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or customer master keys (CMKs) stored in AWS Key Management Service (AWS KMS). When you use server-side encryption, Amazon S3 encrypts an object before saving it to disk and decrypts it when you download the objects.

There is no change to the encryption of the objects that existed in the bucket before default encryption was enabled.

So for the given use-case, you can continue to use the same S3 buckets without impacting operations.

Incorrect options:

RDS - You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created.

However, because you can encrypt a copy of an unencrypted snapshot, you can effectively add encryption to an unencrypted DB instance. That is, you can create a snapshot of your DB instance, and then create an encrypted copy of that snapshot. You can then restore a DB instance from the encrypted snapshot, and thus you have an encrypted copy of your original DB instance.

EBS - There is no direct way to encrypt an existing unencrypted volume or snapshot, you can encrypt them by creating either a volume or a snapshot. If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default key for EBS encryption.

EFS - You can enable encryption of data at rest when creating an Amazon EFS file system. Once the file system is created, you cannot modify the file system to be unencrypted or vice-versa.

References:

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

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

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

Question 52: **Correct**

You work for a blockchain company and you have a ledger application that is memory intensive. It is exposed in an auto scaling group behind a load balancer. You would like to auto scale your application based on the number of users that you have.

As a SysOps Administrator, which of the following would you recommend to meet this requirement?

◯ **Push the RAM usage as a custom metric for the Load Balancer and auto scale based on that**

○ **Deploy a script on the Load Balancer to expose the number of users that are connected to your application as a custom CloudWatch metric**

○ **Auto Scale based on the number of connections CloudWatch metric for the Load Balancer** **(Correct)**

○ **Use the RAM usage CloudWatch metric directly from the Load Balancer and auto scale based on that**

Explanation

Correct option:

Auto Scale based on the number of connections CloudWatch metric for the Load Balancer

Elastic Load Balancing publishes metrics to Amazon CloudWatch for your load balancers and your targets. Elastic Load Balancing reports metrics to CloudWatch only when requests are flowing through the load balancer. If there are requests flowing through the load balancer, Elastic Load Balancing measures and sends its metrics in 60-second intervals. If there are no requests flowing through the load balancer or no data for a metric, the metric is not reported.

For the given use-case, you can use the `ActiveConnectionCount` metric to auto scale. This metric represents the total number of concurrent TCP connections active from clients to the load balancer and from the load balancer to targets.

Incorrect options:

Push the RAM usage as a custom metric for the Load Balancer and auto scale based on that

Use the RAM usage CloudWatch metric directly from the Load Balancer and auto scale based on that

Both these options advocating RAM usage as a metric are incorrect since the use-case refers to the auto scaling criteria based on the number of users which is better represented by the `ActiveConnectionCount` metric.

Deploy a script on the Load Balancer to expose the number of users that are connected to your application as a custom CloudWatch metric - This option has been added as a distractor as you cannot deploy a script on a Load Balancer.

Reference:

https://docs.amazonaws.cn/en_us/elasticloadbalancing/latest/application/load-balancer-cloudwatch-metrics.html

Question 53: **Correct**

You are deploying an application and use the cfn-init and cfn-signal script to ensure the application is properly deployed before signaling to CloudFormation the success of your stack deployment. Right now, every time you deploy, CloudFormation completes successfully, even though the instance is still executing the cfn-init script.

As a SysOps Administrator, which of the following would you identify as the root cause behind the issue?

- ◯ You forgot to include the cfn-signal command in your user data
- ◯ You forgot the Wait Condition **(Correct)**
- ◯ You forgot to include a deletion policy
- ◯ You did not disable Rollbacks

Explanation

Correct option:

You forgot the Wait Condition

The cfn-init helper script reads template metadata from the AWS::CloudFormation::Init key and acts accordingly to:

Fetch and parse metadata from AWS CloudFormation

Install packages

Write files to disk

Enable/disable and start/stop services

The cfn-signal helper script signals AWS CloudFormation to indicate whether Amazon EC2 instances have been successfully created or updated. If you install and configure software applications on instances, you can signal AWS CloudFormation when those software applications are ready.

You can use the wait condition handle to make AWS CloudFormation pause the creation of a stack and wait for a signal before it continues to create the stack. For example, you might want to download and configure applications on an Amazon EC2 instance before considering the creation of that Amazon EC2 instance complete.

AWS CloudFormation creates a wait condition just like any other resource. When AWS CloudFormation creates a wait condition, it reports the wait condition's status as CREATE_IN_PROGRESS and waits until it receives the requisite number of success signals or the wait condition's timeout period has expired. If AWS CloudFormation receives the requisite number of success signals before the time out period expires, it continues creating the stack; otherwise, it sets the wait condition's status to CREATE_FAILED and rolls the stack back.

Incorrect options:

You did not disable Rollbacks - Enabling/disabling rollbacks has no impact on the ability to track the status of the cfn-init script.

You forgot to include the cfn-signal command in your user data - This is a distractor as the cfn-signal command is managed via CloudFormation and not via the user data.

You forgot to include a deletion policy - This is again a distractor as a deletion policy has nothing to do with tracking the status of the cfn-init script.

References:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-helper-scripts-reference.html

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-waitcondition.html

Question 54: **Correct**

Your company has recently been attacked by a team of hackers, exploiting a vulnerability in your Windows OS. A new Windows patch has been released and it needs to be applied as soon as possible to all your instances.

How can you do it?

- ○ **Deploy it using Amazon Inspector**
- ○ **Deploy the patch using Systems Manager** **(Correct)**
- ○ **Use Artifact**
- ○ **Patch the instances directly from the AWS Config interface**

Explanation

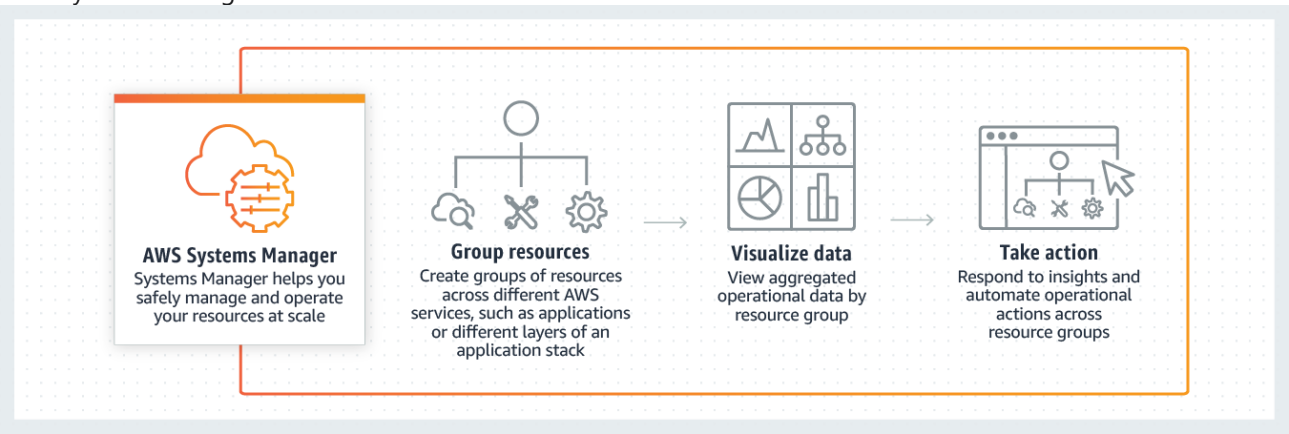
Correct option:

Deploy the patch using Systems Manager

AWS Systems Manager gives you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks such as running commands, managing patches, and configuring servers across AWS Cloud as well as on-premises infrastructure.

With Systems Manager, you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources.

How Systems Manager Works:



via - https://aws.amazon.com/systems-manager/

AWS Systems Manager patch manager helps you select and deploy operating system and software patches automatically across large groups of Amazon EC2 or on-premises instances. Through patch baselines, you can set rules to auto-approve select categories of patches to be installed, such as operating system or high severity patches, and you can specify a list of patches that override these rules and are automatically approved or rejected.

For the given use-case, you can deploy the patches using Systems Manager which supports the patching of both Windows-based and Linux-based instances.

Incorrect options:

Use Artifact - AWS Artifact is a self-service audit artifact retrieval portal that provides our customers with on-demand access to AWS' compliance documentation and AWS agreements. You cannot use Artifact to deploy the patches on the instance.

Deploy it using Amazon Inspector - Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. You cannot use Inspector to deploy the patches on the instance.

Patch the instances directly from the AWS Config interface - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?". You cannot use Config to deploy the patches on the instance.

Reference:

https://aws.amazon.com/systems-manager/faq/

Question 55: **Correct**

Your home-cooking website stores its recipes and comments from users in a Multi-AZ RDS database, which is located in a private subnet. As of yesterday, it seems that your users are unable to access the website and see an error message "512 - Cannot connect to the database".

What could be the reason why the website cannot connect to the database anymore? (Select three)

Network ACL inbound rules have changed **(Correct)**

A read replica has been created recently

Security Group outbound rules have changed

Network ACL outbound rules have changed (Correct)

DB Security Group inbound rules have changed (Correct)

The primary database's private IP has changed

Explanation

Correct options:

DB Security Group inbound rules have changed

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

The following are the characteristics of security group rules:

By default, security groups allow all outbound traffic.

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

Security groups are stateful

For the given use-case, if the DB Security Group inbound rules have changed, then the website may not be able to connect to the database.

Network ACL inbound rules have changed

Network ACL outbound rules have changed

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

The following are the basic things that you need to know about network ACLs:

The default NACL allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.

You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.

You can associate a network ACL with multiple subnets. However, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.

A network ACL contains a numbered list of rules. AWS evaluates the rules in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. AWS recommends that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.

A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.

Network ACLs are stateless, which means that responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

For the given use-case, if the inbound or the outbound NACL rules have changed, then the website may not be able to connect to the database.

Incorrect options:

Security Group outbound rules have changed - Since security groups are stateful so incoming connections can send a reply back to, regardless of any outbound rules, so this option is incorrect.

The primary database's private IP has changed - The private IP associated with an RDS database can change due to things such as multi-AZ failover, so you should use the DNS endpoint to connect to the database. Even if the private IP of the database changes, the DNS endpoint would not change on its own. So this option is incorrect.

A read replica has been created recently - Adding a read replica does not change the primary database's DNS name, so this option is ruled out.

References:

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

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html

Question 56: **Correct**

You host a forum for law questions and per your country's law, you must store all the archives of conversations (about 1 TB) every week for 7 years. These archives must not be tampered with in any way, and you must prove you have set enough controls around your data protection.

What should you do?

- ○ **Store the archives in Glacier and set up a Vault Lock Policy for WORM access** **(Correct)**
- ○ **Store the archives in AWS Artifact and enable compliance monitoring**
- ○ **Store the archives in EBS and use Linux file system protection on the files**
- ○ **Store the archives in S3 and set up a bucket policy, enable versioning and MFA-Delete**

Explanation

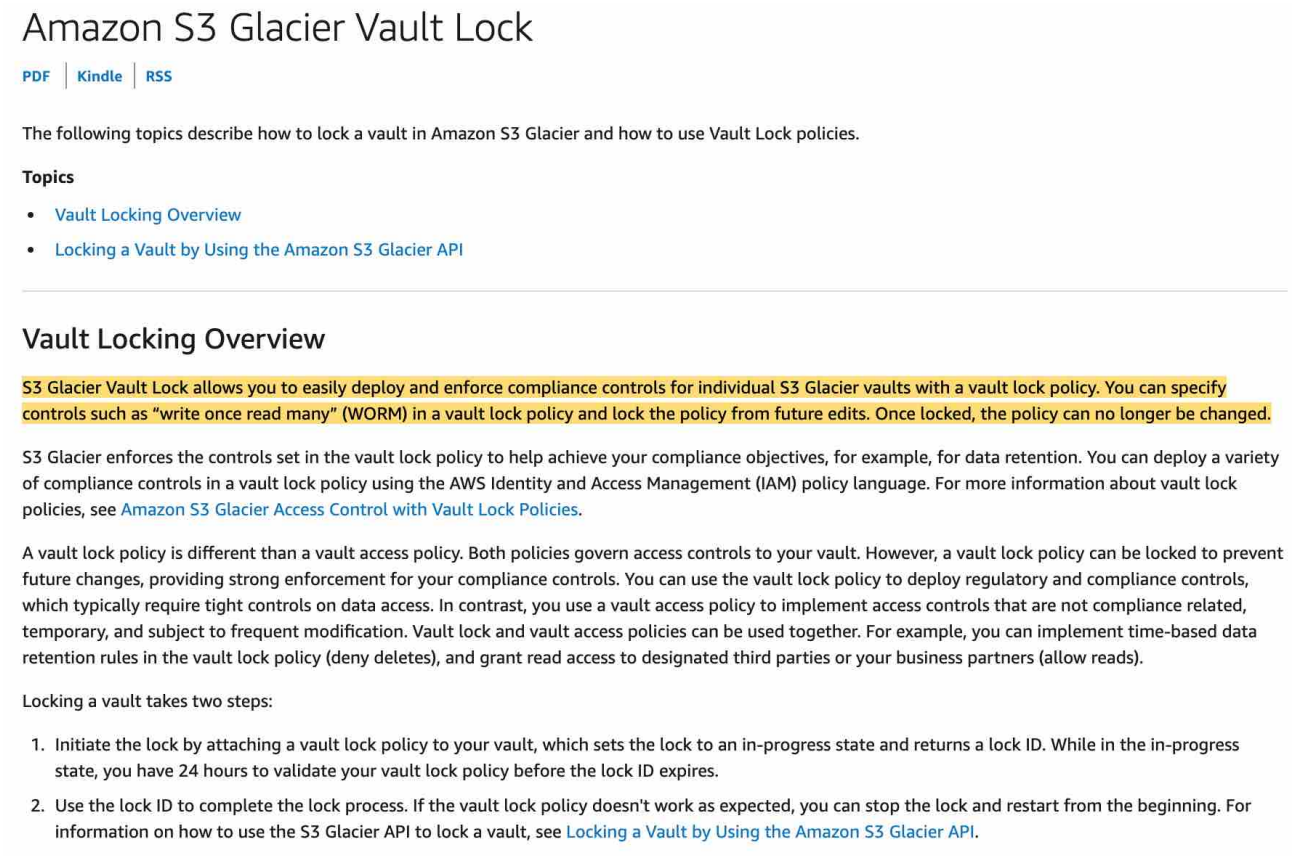
Correct option:

Store the archives in Glacier and set up a Vault Lock Policy for WORM access

You store your data in Amazon S3 Glacier as archives. Archives may be further grouped into vaults.

S3 Glacier Vault Lock allows you to easily deploy and enforce compliance controls for individual S3 Glacier vaults with a vault lock policy. You can specify controls such as "write once read many" (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

For the given use-case, as you need to ensure that the archives are not tampered, so you need to store the archives in Glacier and set up a Vault Lock Policy for WORM access.



Amazon S3 Glacier Vault Lock

PDF | Kindle | RSS

The following topics describe how to lock a vault in Amazon S3 Glacier and how to use Vault Lock policies.

Topics

- Vault Locking Overview
- Locking a Vault by Using the Amazon S3 Glacier API

Vault Locking Overview

S3 Glacier Vault Lock allows you to easily deploy and enforce compliance controls for individual S3 Glacier vaults with a vault lock policy. You can specify controls such as "write once read many" (WORM) in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.

S3 Glacier enforces the controls set in the vault lock policy to help achieve your compliance objectives, for example, for data retention. You can deploy a variety of compliance controls in a vault lock policy using the AWS Identity and Access Management (IAM) policy language. For more information about vault lock policies, see Amazon S3 Glacier Access Control with Vault Lock Policies.

A vault lock policy is different than a vault access policy. Both policies govern access controls to your vault. However, a vault lock policy can be locked to prevent future changes, providing strong enforcement for your compliance controls. You can use the vault lock policy to deploy regulatory and compliance controls, which typically require tight controls on data access. In contrast, you use a vault access policy to implement access controls that are not compliance related, temporary, and subject to frequent modification. Vault lock and vault access policies can be used together. For example, you can implement time-based data retention rules in the vault lock policy (deny deletes), and grant read access to designated third parties or your business partners (allow reads).

Locking a vault takes two steps:

1. Initiate the lock by attaching a vault lock policy to your vault, which sets the lock to an in-progress state and returns a lock ID. While in the in-progress state, you have 24 hours to validate your vault lock policy before the lock ID expires.
2. Use the lock ID to complete the lock process. If the vault lock policy doesn't work as expected, you can stop the lock and restart from the beginning. For information on how to use the S3 Glacier API to lock a vault, see Locking a Vault by Using the Amazon S3 Glacier API.

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

Incorrect options:

Store the archives in S3 and set up a bucket policy, enable versioning and MFA-Delete - Even if you store the archives in a versioned S3 bucket, someone could overwrite the archive and create a new version of it so it does not strictly meet the requirements of the given use-case wherein an existing object mutates into a new version. MFA-Delete would only protect against permanent delete of any object.

Store the archives in EBS and use Linux file system protection on the files Linux file system protection would not be able to enforce compliance controls for the archives in EFS.

Store the archives in AWS Artifact and enable compliance monitoring - AWS Artifact is a self-service audit artifact retrieval portal that provides our customers with on-demand access to AWS' compliance documentation and AWS agreements. You cannot use AWS Artifact to enforce compliance controls for the archives.

Reference:

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

Question 57: **Correct**

Your gp2 drive of 8TB is reaching its peak performance of 10,000 IOPS while being almost fully utilized.

How can you increase the performance while keeping the costs at the same level?

○ **Create two 4 TB gp2 drives and mount them in RAID 0 on the EC2 instance** **(Correct)**

○ **Create two 4 TB gp2 drives and mount them in RAID 1 on the EC2 instance**

◯ Convert the gp2 drive to io1 and increase the PIOPS

◯ Enable burst mode on the gp2 drive

Explanation

Correct option:

Create two 4 TB gp2 drives and mount them in RAID 0 on the EC2 instance

With Amazon EBS, you can use any of the standard RAID configurations that you can use with a traditional bare metal server, as long as that particular RAID configuration is supported by the operating system for your instance. This is because all RAID is accomplished at the software level.

For greater I/O performance than you can achieve with a single volume, RAID 0 can stripe multiple volumes together; for on-instance redundancy, RAID 1 can mirror two volumes together. So for the given use-case, to increase the performance, you should use RAID 0.

RAID Configuration Options

The following table compares the common RAID 0 and RAID 1 options.

Configuration	Use	Advantages	Disadvantages
RAID 0	When I/O performance is more important than fault tolerance; for example, as in a heavily used database (where data replication is already set up separately).	I/O is distributed across the volumes in a stripe. If you add a volume, you get the straight addition of throughput and IOPS.	Performance of the stripe is limited to the worst performing volume in the set. Loss of a single volume results in a complete data loss for the array.
RAID 1	When fault tolerance is more important than I/O performance; for example, as in a critical application.	Safer from the standpoint of data durability.	Does not provide a write performance improvement; requires more Amazon EC2 to Amazon EBS bandwidth than non-RAID configurations because the data is written to multiple volumes simultaneously.

Important

RAID 5 and RAID 6 are not recommended for Amazon EBS because the parity write operations of these RAID modes consume some of the IOPS available to your volumes. Depending on the configuration of your RAID array, these RAID modes provide 20-30% fewer usable IOPS than a RAID 0 configuration. Increased cost is a factor with these RAID modes as well; when using identical volume sizes and speeds, a 2-volume RAID 0 array can outperform a 4-volume RAID 6 array that costs twice as much.

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

Incorrect options:

Convert the gp2 drive to io1 and increase the PIOPS - Changing the gp2 drive to io1 entails more costs as the pricing is $0.10 per GB-month of provisioned storage for gp2 and $0.125 per GB-month of provisioned storage for io1. So this option is ruled out.

Create two 4 TB gp2 drives and mount them in RAID 1 on the EC2 instance - You should use RAID 1 when fault tolerance is more important than I/O performance.

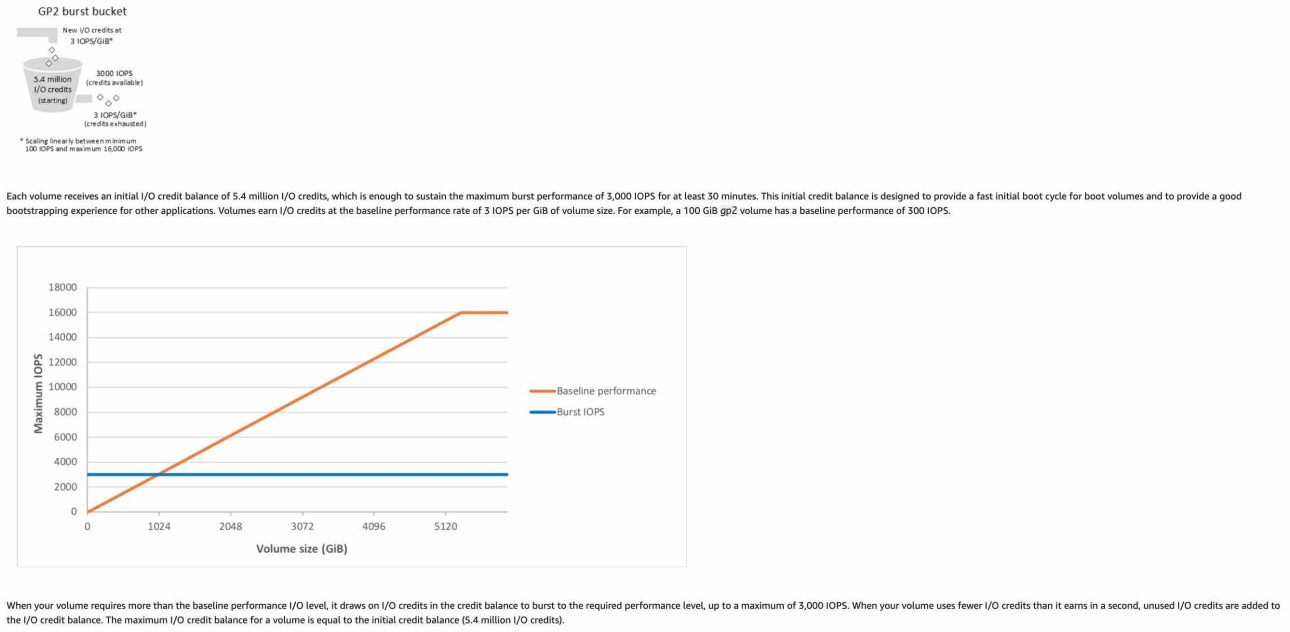
Enable burst mode on the gp2 drive - gp2 volumes can burst to 3,000 IOPS for extended periods of time. This option is a distractor as you do not need to enable the burst mode for gp2 volumes as it's available by default.

General Purpose SSD volumes (gp2)

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver their provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

I/O Credits and burst performance

The performance of gp2 volumes is tied to volume size, which determines the baseline performance level of the volume and how quickly it accumulates I/O credits; larger volumes have higher baseline performance levels and accumulate I/O credits faster. I/O credits represent the available bandwidth that your gp2 volume can use to burst large amounts of I/O when more than the baseline performance is needed. The more credits your volume has for I/O, the more time it can burst beyond its baseline performance level and the better it performs when more performance is needed. The following diagram shows the burst-bucket behavior for gp2.



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

References:

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/raid-config.html

https://aws.amazon.com/blogs/database/understanding-burst-vs-baseline-performance-with-amazon-rds-and-gp2/

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

Question 58: **Correct**

After enabling S3 MFA-Delete, for which actions do you need MFA? (Select two)

Uploading a new object version

Permanently delete an object version (Correct)

Enabling Versioning

Listing deleted versions

Suspending versioning (Correct)

Explanation

Correct options:

Permanently delete an object version

Suspending versioning

You may add another layer of security by configuring a bucket to enable MFA (multi-factor authentication) Delete, which requires additional authentication for either of the following operations:

Change the versioning state of your bucket

Permanently delete an object version

MFA Delete requires two forms of authentication together:

Your security credentials

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

If a bucket's versioning configuration is MFA Delete–enabled, the bucket owner must include the x-amz-mfa request header in requests to permanently delete an object version or change the versioning state of the bucket. Requests that include x-amz-mfa must use HTTPS.

Incorrect options:

Enabling Versioning - You do not need MFA to enable versioning for a bucket.

Listing deleted versions - You do not need MFA to list deleted versions.

Uploading a new object version - You do not need MFA to upload a new object version.

References:

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

Question 59: **Correct**

You have a production Postgres RDS database and a custom rule in AWS Config has been set up and shows that some connections established to your database are not encrypted.

How can you ensure all connections to RDS are encrypted?

- ○ **Patch the database with the SSL/TLS Postgres Addon**
- ○ **Enable SSL connections from the RDS Console**
- ○ **Review the DB parameter groups** **(Correct)**
- ○ **Edit the security group rules**

Explanation

Correct option:

Review the DB parameter groups

You can allow only SSL connections to your RDS for PostgreSQL database instance by enabling the rds.force_ssl parameter ("0" by default) through the parameter groups page on the RDS Console or through the CLI.

Requiring an SSL connection to a PostgreSQL DB instance

You can require that connections to your PostgreSQL DB instance use SSL by using the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to 0 (off). You can set the `rds.force_ssl` parameter to 1 (on) to require SSL for connections to your DB instance. Updating the `rds.force_ssl` parameter also sets the PostgreSQL `ssl` parameter to 1 (on) and modifies your DB instance's `pg_hba.conf` file to support the new SSL configuration.

You can set the `rds.force_ssl` parameter value by updating the parameter group for your DB instance. If the parameter group for your DB instance isn't the default one, and the `ssl` parameter is already set to 1 when you set `rds.force_ssl` to 1, you don't need to reboot your DB instance. Otherwise, you must reboot your DB instance for the change to take effect. For more information on parameter groups, see Working with DB parameter groups.

When the `rds.force_ssl` parameter is set to 1 for a DB instance, you see output similar to the following when you connect, indicating that SSL is now required:

```
$ psql postgres -h SOMEHOST.amazonaws.com -p 8192 -U someuser
. . .
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

Determining the SSL connection status

The encrypted status of your connection is shown in the logon banner when you connect to the DB instance:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=>
```

via
- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html#PostgreSQL.Concepts.General.SSL

Incorrect options:

Edit the security group rules - Security group can be used to allow connections based on certain inbound rules from selected sources. However, you need to use parameter groups to enforce SSL.

Enable SSL connections from the RDS Console - This is a made-up option and has been added as a distractor.

Patch the database with the SSL/TLS Postgres Addon - You cannot install patches on RDS databases as these are completely managed by AWS, so this option is incorrect.

Reference:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html#PostgreSQL.Concepts.General.SSL

Question 60: **Correct**

A project has an Application Load Balancer configured to route each request independently to the registered targets based on the chosen load-balancing algorithm. The team wants to set up a solution that allows the servers to maintain state information to provide a continuous experience to the end-users.

As a SysOps Administrator, which of the following will you identify as the correct way to configure the required session affinity?

◯ **Enable Cookies on all the Application Load Balancer targets**

○ **Enable `X-Forwarded-For` header which can be used to store cookies on the server**

○ **Enable Stickiness on the Application Load Balancer** **(Correct)**

○ **Enable Connection Draining on the Application Load Balancer**

Explanation

Correct option:

Enable Stickiness on the Application Load Balancer - By default, an Application Load Balancer routes each request independently to a registered target based on the chosen load-balancing algorithm. However, you can use the sticky session feature (also known as session affinity) to enable the load balancer to bind a user's session to a specific target. This ensures that all requests from the user during the session are sent to the same target. This feature is useful for servers that maintain state information in order to provide a continuous experience to clients. To use sticky sessions, the client must support cookies.

Application Load Balancers support both duration-based cookies and application-based cookies. The key to managing sticky sessions is determining how long your load balancer should consistently route the user's request to the same target. Sticky sessions are enabled at the target group level. You can use a combination of duration-based stickiness, application-based stickiness, and no stickiness across all of your target groups.

Incorrect options:

Enable Cookies on all the Application Load Balancer targets - Cookies are enabled on the client-side and not on the server-side.

Enable Connection Draining on the Application Load Balancer - To ensure that a Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use connection draining. This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy. When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered.

Enable `X-Forwarded-For` header which can be used to store cookies on the server - The `X-Forwarded-For` request header is automatically added and helps you identify the IP address of a client when you use an HTTP or HTTPS load balancer. Because load balancers intercept traffic between clients and servers, your server access logs contain only the IP address of the load balancer. To see the IP address of the client, use the `X-Forwarded-For` request header.

Reference:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/sticky-sessions.html

Question 61: **Correct**

In order to improve the read performance of the files stored in S3, you have decided to deploy it using CloudFront. As part of this deployment, you would like to ensure that only CloudFront is allowed to access the S3 bucket files.

How can you achieve that?

- ○ **Encrypt all your files using a KMS key that only CloudFront can access**
- ○ **Attaching a security group to S3 and CloudFront and only allow incoming traffic from CloudFront using the security group rules**
- ○ **Attaching an IAM role to CloudFront and defining a bucket policy to only allow this role**
- ○ **Using an Origin Access Identity and a bucket policy** **(Correct)**

Explanation

Correct option:

Using an Origin Access Identity and a bucket policy

To restrict access to content that you serve from Amazon S3 buckets, you need to follow these steps:

Create a special CloudFront user called an origin access identity (OAI) and associate it with your distribution.

Configure your S3 bucket permissions so that CloudFront can use the OAI to access the files in your bucket and serve them to your users. Make sure that users can't use a direct URL to the S3 bucket to access a file there.

Example Amazon S3 bucket policy that gives the OAI read access

The following example allows the OAI to read objects in the specified bucket (`s3:GetObject`).

```
{
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::awsexamplebucket/*"
        }
    ]
}
```

Example Amazon S3 bucket policy that gives the OAI read and write access

The following example allows the OAI to read and write objects in the specified bucket (`s3:GetObject` and `s3:PutObject`). This allows viewers to upload files to your Amazon S3 bucket through CloudFront.

```
{
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity EH1HDMB1FH2TC"
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::aws-example-bucket/*"
        }
    ]
}
```

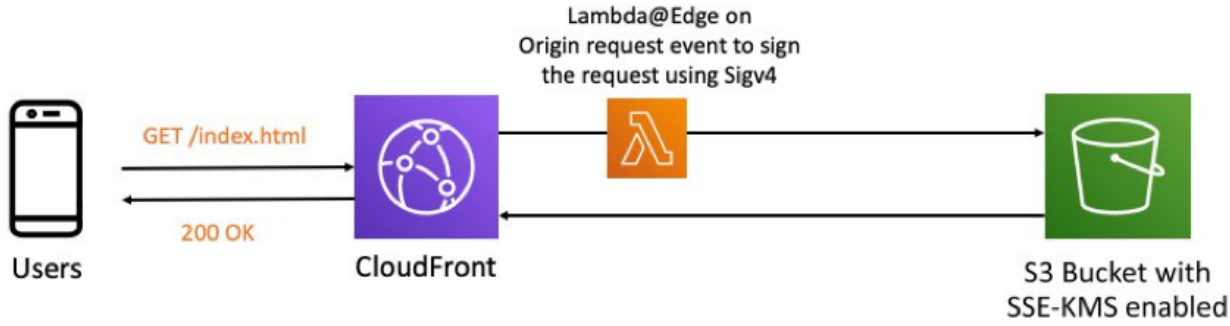
via - https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html

After you take these steps, users can only access your files through CloudFront, not directly from the S3 bucket.

Incorrect options:

Attaching an IAM role to CloudFront and defining a bucket policy to only allow this role - This is a distractor as you cannot associate an IAM role to CloudFront.

Encrypt all your files using a KMS key that only CloudFront can access - Although you could enable SSE-KMS on S3 and serve content using CloudFront by using Lambda@Edge, but this solution does not address the given use-case. You can ensure that only CloudFront is allowed to access the S3 bucket files by using Origin Access Identity and a bucket policy.



via - https://aws.amazon.com/blogs/networking-and-content-delivery/serving-sse-kms-encrypted-content-from-s3-using-cloudfront/

Attaching a security group to S3 and CloudFront and only allow incoming traffic from CloudFront using the security group rules - This is a distractor as you cannot attach a security group to S3 or CloudFront.

References:

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

https://aws.amazon.com/blogs/networking-and-content-delivery/serving-sse-kms-encrypted-content-from-s3-using-cloudfront/

Question 62: **Correct**

You want to improve the process to assign the accounting for your AWS bills to the different departments that the resources belong to. You currently have all your resources under one AWS account.

What is the best way to properly get billing reports for the different company departments, with the least possible administrative overhead?

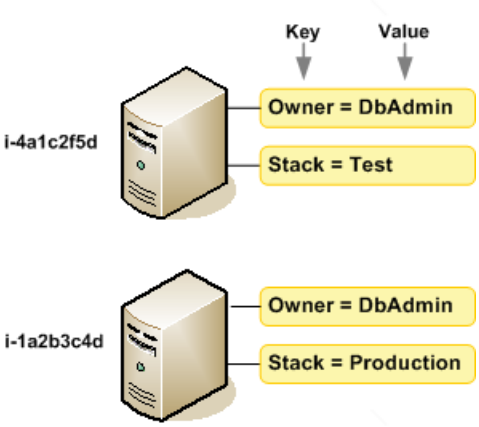
- ◯ **Use AWS Organizations**
- ◯ **Use Tags**
- ◯ **Use Cost Allocation Tags** **(Correct)**
- ◯ **Use EC2 Billing Report**

Explanation

Correct option:

Use Cost Allocation Tags

A tag is a label that you or AWS assigns to an AWS resource. Each tag consists of a key and a value. For each resource, each tag key must be unique, and each tag key can have only one value. You can use tags to organize your resources, and cost allocation tags to track your AWS costs on a detailed level. After you activate cost allocation tags, AWS uses the cost allocation tags to organize your resource costs on your cost allocation report, to make it easier for you to categorize and track your AWS costs.



- https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html

After you or AWS applies tags to your AWS resources (such as Amazon EC2 instances or Amazon S3 buckets) and you activate the tags in the Billing and Cost Management console, AWS generates a cost allocation report as a comma-separated value (CSV file) with your usage and costs grouped by your active tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

At the end of the billing cycle, the total charges (tagged and untagged) on the billing report with cost allocation tags reconciles with the total charges on your Bills page total and other billing reports for the same period.

For the given use-case, you can use cost allocation tags to get billing reports for the different company departments.

Incorrect options:

Use Tags - By default, new tag keys that you add using the API or the AWS Management Console are automatically excluded from the cost allocation report. When you select tag keys to include in your cost allocation report, each key becomes an additional column that lists the value for each corresponding line item. Because you might use tags for more than just your cost allocation report (for example, tags for security or operational reasons), you can include or exclude individual tag keys for the report. By default, you cannot use tags to get billing reports for the different company departments.

Use AWS Organizations - AWS Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts. Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance. You can also simplify billing by setting up a single payment method for all of your AWS accounts. AWS Organizations is available to all AWS customers at no additional charge. You cannot use AWS Organizations to get billing reports for the different company departments.

Use EC2 Billing Report - This is a made-up option as there is no such thing as an EC2 Billing Report.

References:

https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html

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

Question 63: **Incorrect**

A SysOps Administrator has shared an AMI from account A to account B and then de-registered the AMI in a few days.

What is the outcome of this action? (Select two)

You can launch new instances from the de-registered AMI from account B alone. The AMI is invalid in account A

The instances already launched from the shared AMI in account B, are not impacted by this de-registration (Correct)

Copy the AMI to a different Region in account B and create instances from this AMI (Incorrect)

You can't launch new instances from the AMI in account A or in account B (Correct)

Instances launched using the shared AMI in account B are de-registered

Explanation

Correct options:

You can't launch new instances from the AMI in account A or in account B

If you share an AMI from account A to account B, and then deregister the AMI from account A, you can't launch new instances from the AMI in account B.

The instances already launched from the shared AMI in account B, are not impacted by this de-registration - Deregistering the AMI in the original account doesn't impact instances launched from the shared AMI. If you need to launch new instances after deregistering the AMI, you can create a new AMI from one of the new instances.

Incorrect options:

Instances launched using the shared AMI in account B are de-registered - Existing instances are unaffected by changes.

You can launch new instances from the de-registered AMI from account B alone. The AMI is invalid in account A - Instances cannot be launched in either of the accounts with the de-registered AMI.

Copy the AMI to a different Region in account B and create instances from this AMI - This should be done before de-registering the AMI from account A.

Reference:

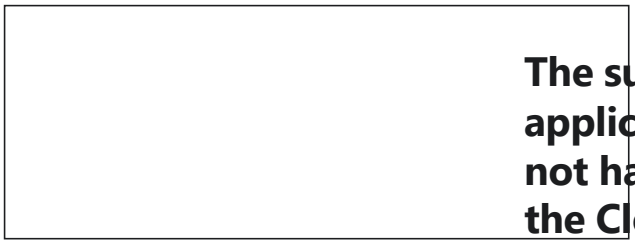
https://aws.amazon.com/premiumsupport/knowledge-center/account-transfer-ec2-instance/

Question 64: **Incorrect**

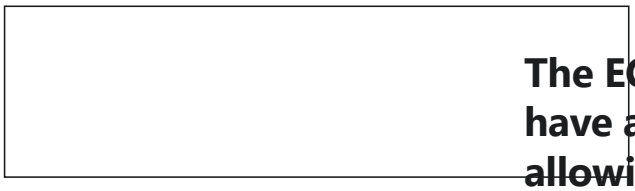
You are provisioning an internal full LAMP stack using CloudFormation, and the EC2 instance gets configured automatically using the cfn helper scripts, such as cfn-init and cfn-signal. The stack creation fails as CloudFormation fails to receive a signal from your EC2 instance.

What are the possible reasons for this? (Select two)

AWS is experiencing an Insufficient Capacity for the instance type you requested



The subnet where the application is deployed does not have a network route to the CloudFormation service through a NAT Gateway or Internet Gateway **(Correct)**



The EC2 instance does not have a proper IAM role allowing to signal the success to CloudFormation **(Incorrect)**

The cfn-signal script does not get executed before the timeout of the wait condition (Correct)

The cfn-init script failed

Explanation

Correct options:

The subnet where the application is deployed does not have a network route to the CloudFormation service through a NAT Gateway or Internet Gateway - As the use-case mentions an internal full LAMP stack, this implies that the stack is to be deployed in a private subnet. Now this private subnet must have a network route to the CloudFormation service through a NAT Gateway or Internet Gateway.

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

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. Internet Gateways must be deployed in a public subnet and the corresponding entry should be added in the route table.

The cfn-signal script does not get executed before the timeout of the wait condition

The Timeout property determines how long AWS CloudFormation waits for the requisite number of success signals. Timeout is a minimum-bound property, meaning the timeout occurs no sooner than the time you specify, but can occur shortly thereafter. The maximum time that you can specify is 43200 seconds (12 hours). For the given scenario, the stack creation can fail as CloudFormation may fail to receive a signal from your EC2 instance if the Timeout property is set to a low value.

Incorrect options:

The EC2 instance does not have a proper IAM role allowing to signal the success to CloudFormation - You do not need an IAM role to use cfn-signal.

AWS is experiencing an Insufficient Capacity for the instance type you requested - In case of Insufficient Capacity, the instance would have not been created and the CloudFormation stack would have failed altogether.

The cfn-init script failed - The cfn-init script failure should still be followed by the cfn-signal script, which would have sent a signal to CloudFormation nonetheless.

References:

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

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

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-waitcondition.html

Question 65: **Correct**

Some of your users' requests are completely being lost due to the metric **SpilloverCount** being greater than 0. This is now happening daily. Your application is running on EC2 instances managed by an ASG running behind a load balancer.

What should you do to prevent this issue from happening?

- ◯ **Monitor for SurgeQueueLength and scale the ASG based on that metric** **(Correct)**
- ◯ **Enable ALB access logs and scale based on CloudWatch Logs**
- ◯ **Pre-warm your load balancer**
- ◯ **Monitor for BackendConnectionErrors and scale the ASG based on that metric**

Explanation

Correct options:

Monitor for SurgeQueueLength and scale the ASG based on that metric

SpilloverCount represents the total number of requests that were rejected because the surge queue is full.

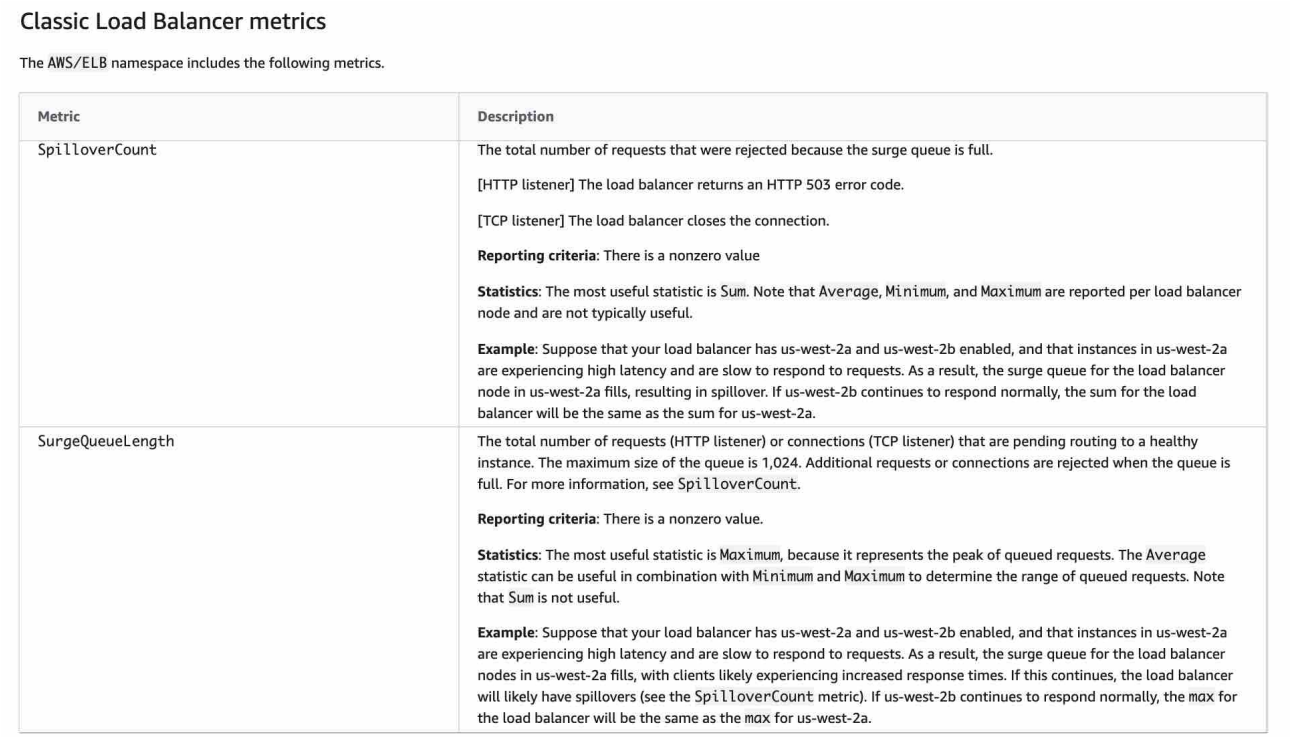
The Classic Load Balancer metric SurgeQueueLength measures the total number of requests queued by your Classic Load Balancer. An increased maximum statistic for SurgeQueueLength indicates that backend systems aren't able to process incoming requests as fast as the requests are received. Possible reasons for a high SurgeQueueLength metric include:

Overloaded Amazon Elastic Compute Cloud (Amazon EC2) instances behind the Classic Load Balancer that are unable to process all incoming requests

Application dependency issues due to external resource performance issues

Maximum allowable connection limits for instances

To solve this use-case, you need to configure the Auto Scaling groups to scale your instances based on the SurgeQueueLength metric.



Classic Load Balancer metrics

The AWS/ELB namespace includes the following metrics.

Metric	Description
SpilloverCount	The total number of requests that were rejected because the surge queue is full. [HTTP listener] The load balancer returns an HTTP 503 error code. [TCP listener] The load balancer closes the connection. **Reporting criteria**: There is a nonzero value **Statistics**: The most useful statistic is Sum. Note that Average, Minimum, and Maximum are reported per load balancer node and are not typically useful. **Example**: Suppose that your load balancer has us-west-2a and us-west-2b enabled, and that instances in us-west-2a are experiencing high latency and are slow to respond to requests. As a result, the surge queue for the load balancer node in us-west-2a fills, resulting in spillover. If us-west-2b continues to respond normally, the sum for the load balancer will be the same as the sum for us-west-2a.
SurgeQueueLength	The total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance. The maximum size of the queue is 1,024. Additional requests or connections are rejected when the queue is full. For more information, see SpilloverCount. **Reporting criteria**: There is a nonzero value. **Statistics**: The most useful statistic is Maximum, because it represents the peak of queued requests. The Average statistic can be useful in combination with Minimum and Maximum to determine the range of queued requests. Note that Sum is not useful. **Example**: Suppose that your load balancer has us-west-2a and us-west-2b enabled, and that instances in us-west-2a are experiencing high latency and are slow to respond to requests. As a result, the surge queue for the load balancer nodes in us-west-2a fills, with clients likely experiencing increased response times. If this continues, the load balancer will likely have spillovers (see the SpilloverCount metric). If us-west-2b continues to respond normally, the max for the load balancer will be the same as the max for us-west-2a.

via - https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-cloudwatch-metrics.html

Incorrect options:

Pre-warm your load balancer - ELB can handle the vast majority of use cases for the customers without requiring "pre-warming" (configuring the load balancer to have the appropriate level of capacity based on expected traffic). In certain scenarios, such as when flash traffic is expected, or in the case where a load test cannot be configured to gradually increase traffic, AWS recommends that you contact AWS to have your load balancer "pre-warmed". AWS will then configure the load balancer to have the appropriate level of capacity based on the traffic that you expect. AWS will need to know the start and end dates of your tests or expected flash traffic, the expected request rate per second, and the total size of the typical request/response that you will be testing.

Here the backend systems aren't able to process incoming requests as fast as the requests are received, so pre-warming your load balancer will not help.

Monitor for BackendConnectionErrors and scale the ASG based on that metric - BackendConnectionErrors represents the number of connections that were not successfully established between the load balancer and the registered instances. Because the load balancer retries the connection when there are errors, this count can exceed the request rate. Note that this count also includes any connection errors related to health checks. Scaling the ASG based on BackendConnectionErrors will not solve the use-case.

Enable ALB access logs and scale based on CloudWatch Logs - This option has been added as a distractor as you cannot scale based on CloudWatch Logs.