A financial analytics company stores their confidential reports in an Amazon S3 bucket. These reports should be preserved for 5 years and these are no more valid or useful for the company after 5 years. Manual deletion is often delayed which results in higher storage costs for the company.

As a SysOps Administrator, which is the simplest solution to delete the expired reports on-time to save costs?

○ **Configure the "Retain Until Date" in the object lock settings to a date that is 5 years from the object creation date and create a lifecycle policy to delete the object 5 years after the object is created** **(Correct)**

○ **Disable versioning on the S3 bucket for which the retention period is being set, to avoid creating retention periods for all versions of the object. Then, configure the retention period in the object lock settings to 5 years**

○ **Configure a lifecycle policy to delete the object 5 years after the object is created**

○ **Configure the Amazon S3 bucket default settings to specify the "Retain Until Date" for all the objects in the bucket**

Explanation

Correct option:

Configure the "Retain Until Date" in the object lock settings to a date that is 5 years from the object creation date and create a lifecycle policy to delete the object 5 years after the object is created - A retention period protects an object version for a fixed amount of time. When you place a retention period on an object version, Amazon S3 stores a timestamp in the object version's metadata to indicate when the retention period expires. After the retention period expires, the object version can be overwritten or deleted unless you also placed a legal hold on the object version. This is where the lifecycle policy kicks in and deletes the object 5 years after the object is created.

You can place a retention period on an object version either explicitly or through a bucket default setting. When you apply a retention period to an object version explicitly, you specify a Retain Until Date for the object version. Amazon S3 stores the Retain Until Date setting in the object version's metadata and protects the object version until the retention period expires.

Incorrect options:

Configure the Amazon S3 bucket default settings to specify the "Retain Until Date" for all the objects in the bucket - When you use bucket default settings, you don't specify a Retain Until Date. Instead, you specify a duration, in either days or years, for which every object version placed in the bucket should be protected.

Disable versioning on the S3 bucket for which the retention period is being set, to avoid creating retention periods for all versions of the object. Then, configure the retention period in the object lock settings to 5 years - This statement is incorrect. Object Lock works only in versioned buckets, and retention periods and legal holds apply to individual object versions. When you lock an object version, Amazon S3 stores the lock information in the metadata for that object version. Placing a retention period or legal hold on an object protects only the version specified in the request.

Configure a lifecycle policy to delete the object 5 years after the object is created - Just using the lifecycle policy will not prevent accidental deletion of the object during the first 5 years of the object, therefore this option is incorrect.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock.html https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock-overview.html

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

Question 2: **Correct**

A healthcare web application has been deployed on Amazon EC2 instances behind an Application Load Balancer (ALB). The application worked well in the development and test environments. In production, however, users are getting logged off and are being asked to log in several times in an hour.

How will you fix this issue and what precaution needs to be taken to avoid recurrence of the issue?

○ **Use Slow Start Mode when registering the targets to ALB. This assures that the instances get enough time to warm up and hence will not lose the cached data**

○ **Enable Sticky Sessions on Application Load Balancer** **(Correct)**

○ **Enable logging on ALB and check the logs to see the error being generated**

○ **Routing configuration of a Load Balancer is used to route traffic to targets. Use this configuration to set the protocol and port number to the correct one**

Explanation

Correct option:

Enable Sticky Sessions on Application Load Balancer - Sticky sessions are a mechanism to route requests to the same target in a target group. This is useful for servers that maintain state information in order to provide a continuous experience to clients. To use sticky sessions, the clients must support cookies.

When a load balancer first receives a request from a client, it routes the request to a target, generates a cookie named AWSALB that encodes information about the selected target, encrypts the cookie, and includes the cookie in the response to the client. The client should include the cookie that it receives in subsequent requests to the load balancer. When the load balancer receives a request from a client that contains the cookie, if sticky sessions are enabled for the target group and the request goes to the same target group, the load balancer detects the cookie and routes the request to the same target. If the cookie is present but cannot be decoded, or if it refers to a target that was deregistered or is unhealthy, the load balancer selects a new target and updates the cookie with information about the new target.

You enable sticky sessions at the target group level. You can also set the duration for the stickiness of the load balancer-generated cookie in seconds. The duration is set with each request. Therefore, if the client sends a request before each duration period expires, the sticky session continues.

Incorrect options:

Use Slow Start Mode when registering the targets to ALB. This assures that the instances get enough time to warm up and hence will not lose the cached data - By default, a target starts to receive its full share of requests as soon as it is registered with a target group and passes an initial health check. Using slow start mode gives targets time to warm up before the load balancer sends them a full share of requests. However, this option cannot be used to address the given use-case.

Routing configuration of a Load Balancer is used to route traffic to targets. Use this configuration to set the protocol and port number to the correct one - By default, a load balancer routes requests to its targets using the protocol and port number that you specified when you created the target group. Alternatively, you can override the port used for routing traffic to a target when you register it with the target group. This, however, has nothing to do with users getting logged off every few minutes.

Enable logging on ALB and check the logs to see the error being generated - The current use case points at issues with session data and hence using sticky sessions is the right answer here.

Reference:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html#sticky-sessions

Question 3: **Correct**

A retail company is working on moving their technology infrastructure to AWS Cloud. The company has developed several custom scripts to monitor the instances hosting their applications and want to reuse these scripts on AWS Cloud. The development team is looking at a way to disable the pre-existing Amazon EC2 status checks.

As a SysOps Administrator, which of the following will you suggest to meet the given requirement?

○ **Status checks are built into Amazon EC2, so they cannot be disabled or deleted** **(Correct)**

○ **Status checks are built into Amazon EC2, so they cannot be disabled, but can be deleted from active configuration**

○ **Amazon EC2 automated checks to identify hardware issues cannot be disabled. Automated checks for software issues can however be disabled**

○ **Amazon EC2 status checks are interwoven into CloudWatch metrics. You can disable EC2 instance status checks from the CloudWatch metrics console**

Explanation

Correct option:

Status checks are built into Amazon EC2, so they cannot be disabled or deleted

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems.

Status checks are performed every minute, returning a pass or a fail status. If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired. Status checks are built into Amazon EC2, so they cannot be disabled or deleted.

When a status check fails, the corresponding CloudWatch metric for status checks is incremented. You can use these metrics to create CloudWatch alarms that are triggered based on the result of the status checks. For example, you can create an alarm to warn you if status checks fail on a specific instance. You can also create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying issue.

Incorrect options:

Status checks are built into Amazon EC2, so they cannot be disabled, but can be deleted from active configuration

Amazon EC2 automated checks to identify hardware issues cannot be disabled. Automated checks for software issues can however be disabled

Amazon EC2 status checks are interwoven into CloudWatch metrics. You can disable EC2 instance status checks from the CloudWatch metrics console

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

Reference:

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

Question 4: **Incorrect**

A retail company stores its business-critical files on an Amazon S3 bucket that is also configured as a website endpoint. The company needs a robust configuration that will allow access only through CloudFront. No user or team member should be able to access the files directly from Amazon S3 URL.

As a SysOps Administrator, which of the following would you suggest to address this requirement?

○ **Setup the Amazon S3 bucket as a custom origin with CloudFront. Restrict the access to content by setting up custom headers** **(Correct)**

○ **Configure a Security Group with CloudFront to restrict access to users**

○ **Create an Origin Access Identity (OAI) and configure S3 bucket permissions so that CloudFront can use the OAI to access the files in your bucket** **(Incorrect)**

○ **Configure a Network Access Control List (ACL) with CloudFront to restrict access to users**

Explanation

Correct option:

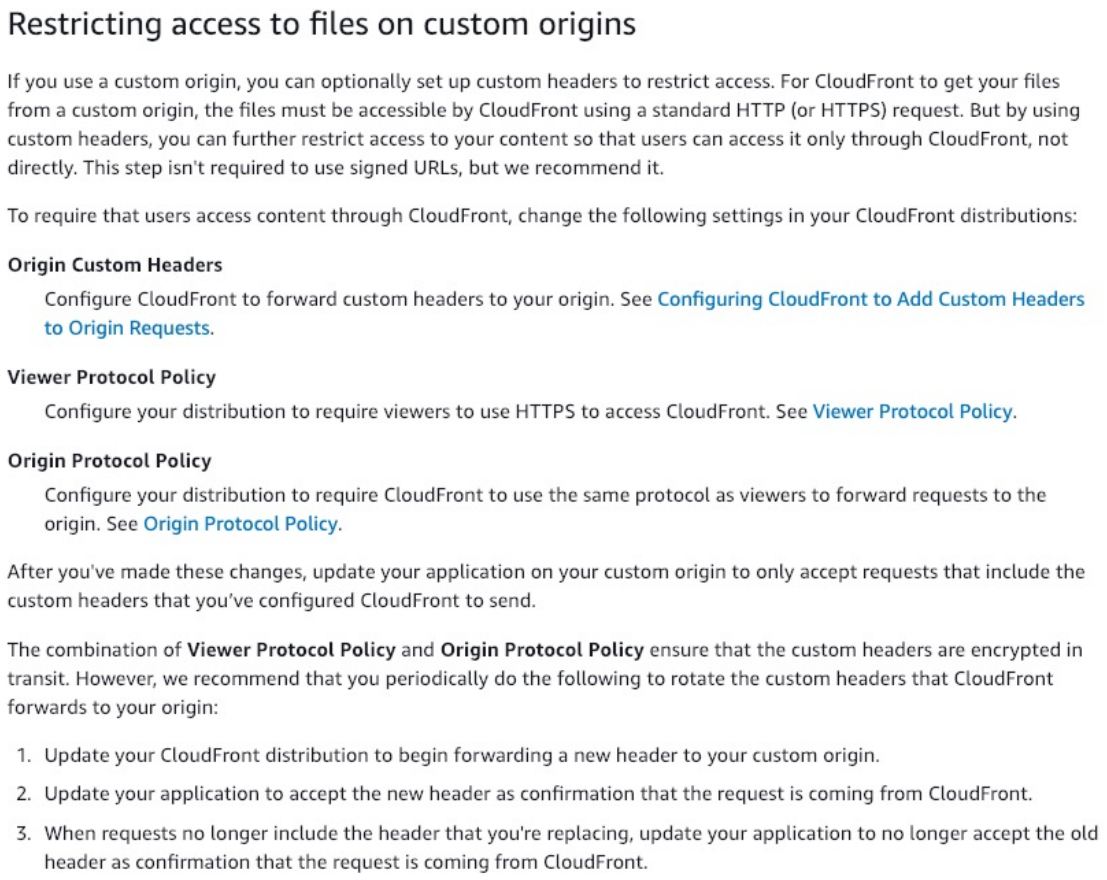
Setup the Amazon S3 bucket as a custom origin with CloudFront. Restrict the access to content by setting up custom headers

If you use an Amazon S3 bucket configured as a website endpoint, you must set it up with CloudFront as a custom origin. You can't use the origin access identity feature. However, you can restrict access to content on a custom origin by setting up custom headers and configuring your origin to require them.

To require that users access content through CloudFront, change the following settings in your CloudFront distributions: 1. Origin Custom Headers: Configure CloudFront to forward custom headers to your origin. 2. Viewer Protocol Policy: Configure your distribution to require viewers to use HTTPS to access CloudFront. 3. Origin Protocol Policy: Configure your distribution to require CloudFront to use the same protocol as viewers to forward requests to the origin.

After you've made these changes, update your application on your custom origin to only accept requests that include the custom headers that you've configured CloudFront to send.

More info on restricting access to files on custom origins:



Restricting access to files on custom origins

If you use a custom origin, you can optionally set up custom headers to restrict access. For CloudFront to get your files from a custom origin, the files must be accessible by CloudFront using a standard HTTP (or HTTPS) request. But by using custom headers, you can further restrict access to your content so that users can access it only through CloudFront, not directly. This step isn't required to use signed URLs, but we recommend it.

To require that users access content through CloudFront, change the following settings in your CloudFront distributions:

Origin Custom Headers

Configure CloudFront to forward custom headers to your origin. See Configuring CloudFront to Add Custom Headers to Origin Requests.

Viewer Protocol Policy

Configure your distribution to require viewers to use HTTPS to access CloudFront. See Viewer Protocol Policy.

Origin Protocol Policy

Configure your distribution to require CloudFront to use the same protocol as viewers to forward requests to the origin. See Origin Protocol Policy.

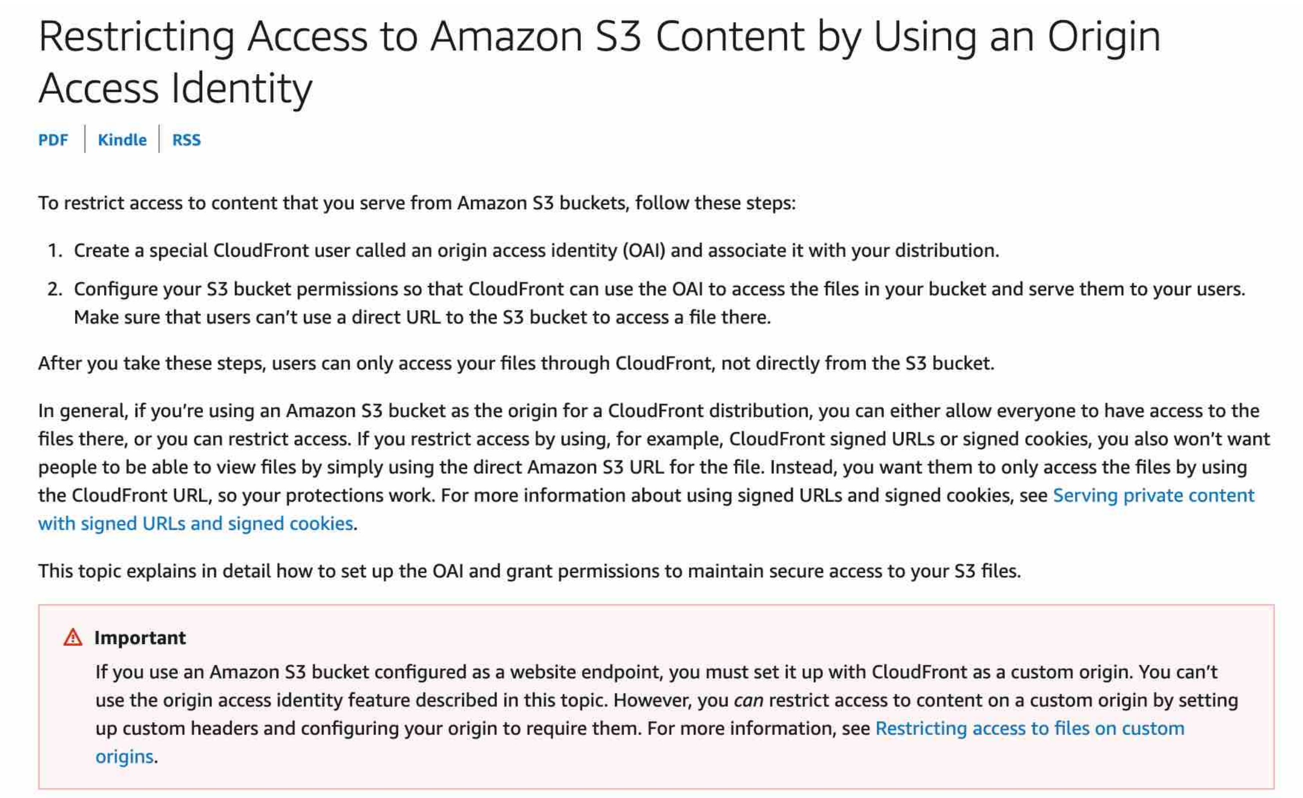
After you've made these changes, update your application on your custom origin to only accept requests that include the custom headers that you've configured CloudFront to send.

The combination of **Viewer Protocol Policy** and **Origin Protocol Policy** ensure that the custom headers are encrypted in transit. However, we recommend that you periodically do the following to rotate the custom headers that CloudFront forwards to your origin:

1. Update your CloudFront distribution to begin forwarding a new header to your custom origin.
2. Update your application to accept the new header as confirmation that the request is coming from CloudFront.
3. When requests no longer include the header that you're replacing, update your application to no longer accept the old header as confirmation that the request is coming from CloudFront.

via

- https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-overview.html#forward-custom-headers-restrict-access



Restricting Access to Amazon S3 Content by Using an Origin Access Identity

PDF | Kindle | RSS

To restrict access to content that you serve from Amazon S3 buckets, follow these steps:

1. Create a special CloudFront user called an origin access identity (OAI) and associate it with your distribution.
2. 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.

After you take these steps, users can only access your files through CloudFront, not directly from the S3 bucket.

In general, if you're using an Amazon S3 bucket as the origin for a CloudFront distribution, you can either allow everyone to have access to the files there, or you can restrict access. If you restrict access by using, for example, CloudFront signed URLs or signed cookies, you also won't want people to be able to view files by simply using the direct Amazon S3 URL for the file. Instead, you want them to only access the files by using the CloudFront URL, so your protections work. For more information about using signed URLs and signed cookies, see Serving private content with signed URLs and signed cookies.

This topic explains in detail how to set up the OAI and grant permissions to maintain secure access to your S3 files.

⚠ **Important**

If you use an Amazon S3 bucket configured as a website endpoint, you must set it up with CloudFront as a custom origin. You can't use the origin access identity feature described in this topic. However, you *can* restrict access to content on a custom origin by setting up custom headers and configuring your origin to require them. For more information, see Restricting access to files on custom origins.

via - https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html

Incorrect options:

Create an Origin Access Identity (OAI) and configure S3 bucket permissions so that CloudFront can use the OAI to access the files in your bucket - As explained above, if you use an Amazon S3 bucket configured as a website endpoint, you can't use the origin access identity feature.

Configure a Security Group with CloudFront to restrict access to users - A Security Group acts as a virtual firewall for Amazon EC2 instances to control incoming and outgoing traffic. Security Groups cannot be used with CloudFront.

Configure a Network Access Control List (ACL) with CloudFront to restrict access to users - 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. Network ACLs are not used with CloudFront.

References:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-overview.html#forward-custom-headers-restrict-access

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/DownloadDistS3AndCustomOrigins.html#concept_S3Origin_website

Question 5: **Incorrect**

A university has just registered for an AWS account to help provide the necessary cloud infrastructure for the students planning to implement a Big Data analytics workflow as part of their thesis. The university has hired you to set up this infrastructure and help their technology team understand the basics of the AWS Virtual Private Cloud (VPC).

As a SysOps Administrator, which of these would you identify as the correct options regarding VPC configurations? (Select three)

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block **(Correct)**

A private subnet, by default, does not have inbound data traffic from the internet. You create a route to the Internet Gateway in the subnet route table to allow access to the private subnet

Subnets, like VPCs, can span across Availability Zones, but will remain in a single AWS Region **(Incorrect)**

Regardless of the type of subnet, the internal IPv4 address range of the subnet is always private **(Correct)**

By default, all subnets can route between each other, whether they are private or public (Correct)

If a subnet has a route to an internet gateway, along with traffic that can be routed to a virtual private gateway for a Site-to-Site VPN connection, the subnet is known as a VPN-only subnet

Explanation

Correct option:

Regardless of the type of subnet, the internal IPv4 address range of the subnet is always private - Regardless of the type of subnet, the internal IPv4 address range of the subnet is always private. AWS never announces these address blocks to the internet.

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block - When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. This is the primary CIDR block for your VPC.

By default, all subnets can route between each other, whether they are private or public - Subnets created in a VPC can communicate with each other. The main route table facilitates this communication.

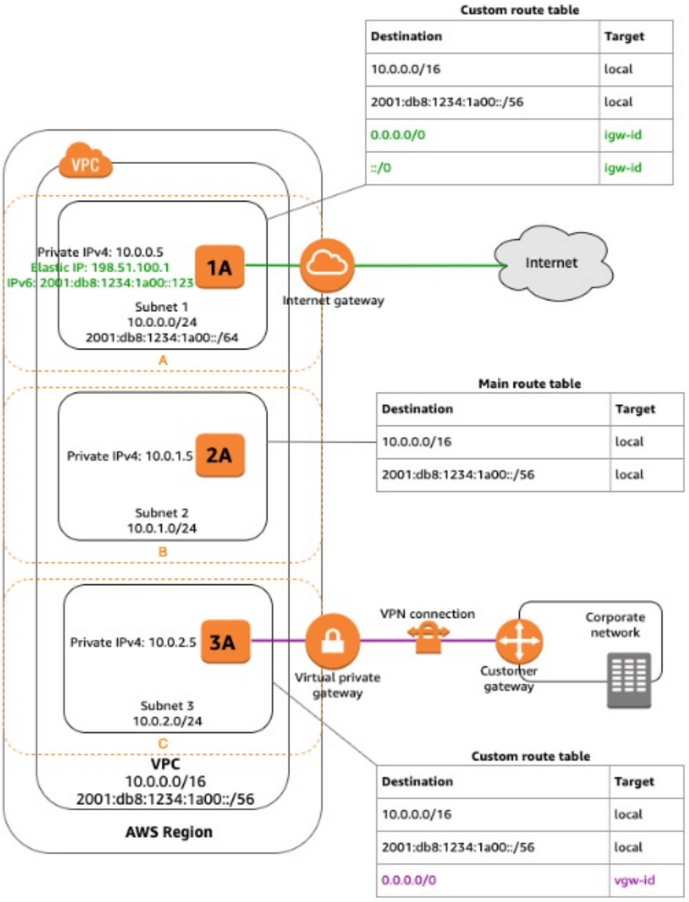
Incorrect options:

Subnets, like VPCs, can span across Availability Zones, but will remain in a single AWS Region - Subnets must reside entirely within one Availability Zone and cannot span across AZs.

A private subnet, by default, does not have inbound data traffic from the internet. You create a route to the Internet Gateway in the subnet route table to allow access to the private subnet - If a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet. A private subnet leverages a public subnet to connect to the internet via the NAT Gateway. A private subnet cannot directly connect to the public internet (In the diagram shown below, subnet 2 is a private subnet).

If a subnet has a route to an internet gateway, along with traffic that can be routed to a virtual private gateway for a Site-to-Site VPN connection, the subnet is known as a VPN-only subnet - If a subnet doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a Site-to-Site VPN connection, the subnet is known as a VPN-only subnet (In the diagram shown below, subnet 3 is a VPN-only subnet).

Example VPC that has been configured with subnets in multiple Availability Zones:



Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	igw-id
::/0	igw-id

Main route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local

Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	vgw-id

via

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

Reference:

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

Question 6: **Incorrect**

A junior developer working on configuring CloudWatch alarms is unable to figure out why a particular CloudWatch Alarm is constantly in the ALARM state.

As a SysOps Administrator, which of these options would you suggest as a fix for the issue?

○ **Once an alarm is triggered and an action is performed, the application logic has to reset the alarm to its normal state. This code has to be included by the development team**

○ **CloudWatch alarm has been incorrectly configured and needs to be deleted and re-configured for fixing the persistent error**

○ **Alarms continue to evaluate metrics against the configured threshold, even after they have already triggered. You can adjust the alarm threshold if you do not want it to be in ALARM state**

(Correct)

○ **Custom alarms once triggered remain in Alarm state till they are manually disabled from either the AWS Console or through application code** **(Incorrect)**

Explanation

Correct option:

Alarms continue to evaluate metrics against the configured threshold, even after they have already triggered. You can adjust the alarm threshold if you do not want it to be in ALARM state

Alarms continue to evaluate metrics against your chosen threshold, even after they have already triggered. This allows you to view its current up-to-date state at any time. You may notice that one of your alarms stays in the ALARM state for a long time. If your metric value is still in breach of your threshold, the alarm will remain in the ALARM state until it no longer breaches the threshold. This is normal behavior. If you want your alarm to treat this new level as OK, you can adjust the alarm threshold accordingly.

Incorrect options:

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

Once an alarm is triggered and an action is performed, the application logic has to reset the alarm to its normal state. This code has to be included by the development team

Custom alarms once triggered remain in Alarm state till they are manually disabled from either the AWS Console or through application code

CloudWatch alarm has been incorrectly configured and needs to be deleted and re-configured for fixing the persistent error

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

Reference:

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

Question 7: **Incorrect**

A systems administrator is configuring Amazon EC2 status check alarm to publish a notification to an SNS topic when the instance fails either the instance check or system status check.

Which CloudWatch metric is the right choice for this configuration?

○ `CombinedStatusCheckFailed`

○ `StatusCheckFailed_System` **(Incorrect)**

○ `StatusCheckFailed_Instance`

○ `StatusCheckFailed` **(Correct)**

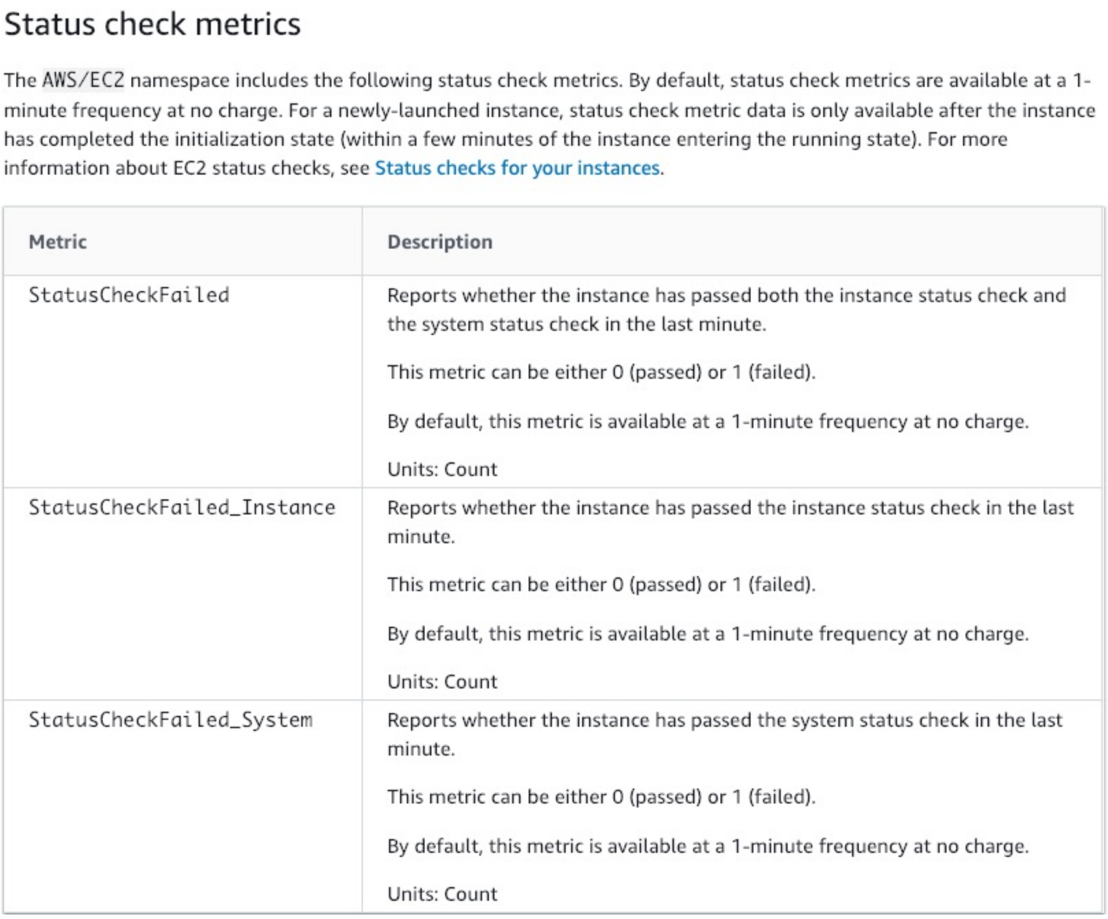
Explanation

Correct option:

`StatusCheckFailed` - The `AWS/EC2` namespace includes a few status check metrics. By default, status check metrics are available at a 1-minute frequency at no charge. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the running state).

`StatusCheckFailed` - Reports whether the instance has passed both the instance status check and the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge.

List of EC2 status check metrics:



Status check metrics

The AWS/EC2 namespace includes the following status check metrics. By default, status check metrics are available at a 1-minute frequency at no charge. For a newly-launched instance, status check metric data is only available after the instance has completed the initialization state (within a few minutes of the instance entering the running state). For more information about EC2 status checks, see Status checks for your instances.

Metric	Description
StatusCheckFailed	Reports whether the instance has passed both the instance status check and the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge. Units: Count
StatusCheckFailed_Instance	Reports whether the instance has passed the instance status check in the last minute. This metric can be either 0 (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge. Units: Count
StatusCheckFailed_System	Reports whether the instance has passed the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed). By default, this metric is available at a 1-minute frequency at no charge. Units: Count

via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html#status-check-metrics

Incorrect options:

`CombinedStatusCheckFailed` - This is a made-up option, given only as a distractor.

`StatusCheckFailed_Instance - Reports whether the instance has passed the instance status check in the last minute. This metric can be either 0 (passed) or 1 (failed).

`StatusCheckFailed_System` - Reports whether the instance has passed the system status check in the last minute. This metric can be either 0 (passed) or 1 (failed).

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html#status-check-metrics

Question 8: **Incorrect**

An hour after launching an important feature on its website, an analytics company has realized that an important page has issues that need to be addressed. The web application is hosted on an Amazon EC2 instance with CloudFront being used to reduce latency for the users. Few users have already accessed this page and the company wants to pull it down as soon as possible.

What should the company do to quickly remove the file from the CloudFront distribution?

○ **By default, CloudFront caches files in edge locations for 24 hours. So, it's not possible to remove the file before this time**

○ **Invalidate the file from CloudFront distribution so that the file is removed immediately** (Correct)

○ **Use CloudFront policies to control what the users can see**

○ **Specify a default root object to show only this object and not the faulty web page** (Incorrect)

Explanation

Correct option:

Invalidate the file from CloudFront distribution so that the file is removed immediately

If you need to remove a file from CloudFront edge caches before it expires, you can do one of the following:

1. Invalidate the file from edge caches. The next time a viewer requests the file, CloudFront returns to the origin to fetch the latest version of the file.
2. Use file versioning to serve a different version of the file that has a different name.

You can use the CloudFront console to create and run an invalidation, display a list of the invalidations that you submitted previously, and display detailed information about an individual invalidation. You can also copy an existing invalidation, edit the list of file paths, and run the edited invalidation. You can't remove invalidations from the list.

When you submit an invalidation request to CloudFront, CloudFront forwards the request to all edge locations within a few seconds, and each edge location starts processing the invalidation immediately. As a result, you can't cancel an invalidation after you submit it.

Incorrect options:

Specify a default root object to show only this object and not the faulty web page - You can configure CloudFront to return a specific object (the default root object) when a user requests the root URL for your web distribution instead of requesting an object in your distribution. Specifying a default root object lets you avoid exposing the contents of your distribution. Although a useful feature, it is not helpful for the given use-case.

By default, CloudFront caches files in edge locations for 24 hours. So, it's not possible to remove the file before this time - It is true that CloudFront caches files in edge locations for 24 hours. But, it's possible to remove them by invalidating the files or versioning the files.

Use CloudFront policies to control what the users can see - With CloudFront policies, you can control the values that are included in the cache key for objects that are cached at CloudFront edge locations. These values can include HTTP request query strings, headers, and cookies.

Reference:

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

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/working-with-policies.html

Question 9: **Incorrect**

A team needs to create an AMI from their Amazon EC2 instances for use in another environment.

What is the right way to create an application-consistent AMI from existing EC2 instances?

○ **Create the AMI by disabling the `No reboot` option** (Correct)

○ **Create an EBS-backed AMI for application consistency** (Incorrect)

○ **Create the AMI with `Delete on termination` enabled**

○ **Create the AMI with `No reboot` option enabled**

Explanation

Correct option:

Create the AMI by disabling the `No reboot` option - On the `Create image` page, `No reboot` flag is present. The default functionality is, Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. When `No reboot` option is selected, the instance is not shut down while creating the AMI. This option is not selected by default.

If you select `No reboot`, the AMI will be crash-consistent (all the volumes are snapshotted at the same time), but not application-consistent (all the operating system buffers are not flushed to disk before the snapshots are created).

Incorrect options:

Create the AMI with `No reboot` option enabled - If the `No reboot` flag is selected, the instance is not shutdown while creating an AMI. This implies, the Operating System buffers are not flushed before creating an AMI, so data integrity could be an issue with AMIs created in this way. Such AMIs are crash-consistent but not application-consistent.

Create an EBS-backed AMI for application consistency - When you create a new instance from an EBS-backed AMI, you are using persistent storage. `No reboot` flag should still be unchecked to ensure that everything on the instance is stopped and in a consistent state during the creation process.

Create the AMI with `Delete on termination` enabled - If you select `Delete on termination`, when you terminate the instance created from this AMI, the EBS volume is deleted. If you clear `Delete on termination`, when you terminate the instance, the EBS volume is not deleted. This option has been added as a distractor.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html

Question 10: **Incorrect**

A SysOps administrator has deployed multiple applications on a fleet of Amazon EC2 instances.

What is the right way to configure scheduled events for these EC2 instances?

○ **Use CloudWatch Agent to configure scheduled events on Amazon EC2 instances**

○ **Use CloudWatch Alarm to configure scheduled events on Amazon EC2 instances**

○ **Use Amazon EventBridge to configure scheduled events on Amazon EC2 instances** **(Incorrect)**

○ **Scheduled events are managed by AWS, you cannot configure scheduled events for your instances** **(Correct)**

Explanation

Correct option:

Scheduled events are managed by AWS, you cannot configure scheduled events for your instances - AWS can schedule events for your instances, such as a reboot, stop/start, or retirement. These events do not occur frequently. If one of your instances will be affected by a scheduled event, AWS sends an email to the email address that's associated with your AWS account before the scheduled event. The email provides details about the event, including the start and end date. Depending on the event, you might be able to take action to control the timing of the event. AWS also sends an AWS Health event, which you can monitor and manage using Amazon CloudWatch Events.

Scheduled events are managed by AWS; you cannot schedule events for your instances. You can view the events scheduled by AWS, customize scheduled event notifications to include or remove tags from the email notification, perform actions when an instance is scheduled to reboot, retire, or stop.

Incorrect options:

Use CloudWatch Alarm to configure scheduled events on Amazon EC2 instances - CloudWatch alarms can be used to watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. CloudWatch Alarm cannot be used to configure scheduled events on Amazon EC2 instances.

Use Amazon EventBridge to configure scheduled events on Amazon EC2 instances - Amazon EventBridge helps automate your AWS services and respond automatically to system events. Events from AWS services are delivered to EventBridge in near real-time, and you can specify automated actions to take when an event matches a rule you write. EventBridge cannot be used to configure scheduled events on Amazon EC2 instances.

Use CloudWatch Agent to configure scheduled events on Amazon EC2 instances - CloudWatch Agent helps collect logs and system-level metrics from both hosts and guests on your EC2 instances and on-premises servers. CloudWatch Agent cannot be used to configure scheduled events on Amazon EC2 instances.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-instances-status-check_sched.html

Question 11: **Correct**

An IT services company runs its technology infrastructure on AWS Cloud. The company runs audits for all the development and testing teams against the standards set by the organization. During a recent audit, the company realized that most of the patch compliance standards are not being followed by the teams. The teams have however tagged all their AWS resources as per the guidelines.

As a SysOps Administrator, which of the following would you recommend as an easy way of fixing the issue as quickly as possible?

○ **Use AWS Systems Manager Patch Manager to automate the process of patching managed instances** **(Correct)**

○ **Use AWS Systems Manager Automation to simplify the patch application process across all instances**

○ **Use Amazon Inspector to automate the process of patching instances that helps improve the security and compliance of the instances**

○ **Use Amazon Patch Manager to automate the process of patching instances**

Explanation

Correct option:

Use AWS Systems Manager Patch Manager to automate the process of patching managed instances

AWS Systems Manager Patch 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. 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 EC2 instances or your on-premises servers and virtual machines (VMs) by operating system type.

Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches. You can install patches on a regular basis by scheduling patching to run as a Systems Manager maintenance window task. You can also install patches individually or to large groups of instances by using Amazon EC2 tags. (Tags are keys that help identify and sort your resources within your organization.) You can add tags to your patch baselines themselves when you create or update them.

Patch Manager provides options to scan your instances and report compliance on a schedule, install available patches on a schedule, and patch or scan instances on demand whenever you need to.

Patch Manager integrates with AWS Identity and Access Management (IAM), AWS CloudTrail, and Amazon EventBridge to provide a secure patching experience that includes event notifications and the ability to audit usage.

Incorrect options:

Use Amazon Inspector to automate the process of patching instances that helps improve the security and compliance of the instances - Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. Inspector is not a patch management service.

Use Amazon Patch Manager to automate the process of patching instances - This is a made-up option and given only as a distractor.

Use AWS Systems Manager Automation to simplify the patch application process across all instances - Systems Manager Automation simplifies common maintenance and deployment tasks of EC2 instances and other AWS resources. Automation enables you to do the following: Build Automation workflows to configure and manage instances and AWS resources, Create custom workflows or use pre-defined workflows maintained by AWS, Receive notifications about Automation tasks and workflows by using Amazon EventBridge, Monitor Automation progress and execution details by using the Amazon EC2 or the AWS Systems Manager console. Systems Manager Automation, however, does not include patch management.

References:

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

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

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

Question 12: **Correct**

Your company has decided that certain users should have Multi-Factor Authentication (MFA) enabled for their sign-in credentials. A newly hired manager has a Gemalto MFA device that he used in his earlier company. He has approached you to configure it for his AWS account.

How will you configure his existing Gemalto MFA device so he can seamlessly connect with AWS services in the new company?

○ **Security constraints mandate that sharing of secrets between multiple parties can only happen in edge cases. Hence, formal approval is needed between AWS and the previous company to use the same Gemalto device**

○ **AWS MFA relies on knowing a unique secret associated with your hardware MFA. This has to be generated again with AWS MFA for the Gemalto device to work with AWS**

○ **AWS MFA does not support the use of your existing Gemalto device** **(Correct)**

○ **You can re-use an existing Gemalto device with AWS MFA, as Gemalto devices do not share any secrets between multiple parties**

Explanation

Correct option:

AWS MFA does not support the use of your existing Gemalto device - AWS MFA relies on knowing a unique secret associated with your hardware MFA (Gemalto) device in order to support its use. Because of security constraints that mandate such secrets never be shared between multiple parties, AWS MFA cannot support the use of your existing Gemalto device. Only a compatible hardware MFA device purchased from Gemalto can be used with AWS MFA. You can re-use an existing U2F security key with AWS MFA, as U2F security keys do not share any secrets between multiple parties.

Incorrect options:

You can re-use an existing Gemalto device with AWS MFA, as Gemalto devices do not share any secrets between multiple parties - As discussed above, you cannot re-use an existing Gemalto device with AWS MFA because secrets cannot be shared with multiple parties.

AWS MFA relies on knowing a unique secret associated with your hardware MFA. This has to be generated again with AWS MFA for the Gemalto device to work with AWS - As discussed above, an existing Gemalto device cannot be used with AWS MFA.

Security constraints mandate that sharing of secrets between multiple parties can only happen in edge cases. Hence, formal approval is needed between AWS and the previous company to use the same Gemalto device - This is a made-up option, given only as a distractor.

Reference:

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

Question 13: **Correct**

As a SysOps Administator, you are writing a CloudFormation template in YAML. The template consists of an EC2 instance creation and one RDS resource. Once your resources are created you would like to output the connection endpoint for the RDS database.

Which intrinsic function returns the value needed?

○ `!FindInMap`

○ `!Sub`

○ `!Ref`

○ `!GetAtt` **(Correct)**

Explanation

Correct option:

AWS CloudFormation provides several built-in functions that help you manage your stacks. Intrinsic functions are used in templates to assign values to properties that are not available until runtime.

`!GetAtt` - The Fn::GetAtt intrinsic function returns the value of an attribute from a resource in the template. This example snippet returns a string containing the DNS name of the load balancer with the logical name myELB - YML : !GetAtt myELB.DNSName JSON : "Fn::GetAtt" : ["myELB" , "DNSName"]

Incorrect options:

`!Sub` - The intrinsic function Fn::Sub substitutes variables in an input string with values that you specify. In your templates, you can use this function to construct commands or outputs that include values that aren't available until you create or update a stack.

`!Ref` - The intrinsic function Ref returns the value of the specified parameter or resource.

`!FindInMap` - The intrinsic function Fn::FindInMap returns the value corresponding to keys in a two-level map that is declared in the Mappings section. For example, you can use this in the Mappings section that contains a single map, RegionMap, that associates AMIs with AWS regions.

References:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-getatt.html

Question 14: **Correct**

A personal care web application is hosted on Amazon EC2 instance in two different Availability Zones (AZs). The application uses Internet Protocol version 6 (IPv6) for communication. The EC2 instances are placed in private subnets. The instances need Internet access to download software updates twice a month.

Which configuration will help achieve this requirement without exposing the instances to the outside world?

○ **Configure an Internet Gateway to allow outbound communication on IPv6 for the instances in the private subnet for your VPC. Public subnets are by default connected to the internet and do not need any extra configuration**

○ **Configure Egress-only Internet Gateway, that allows outbound communication over IPv6 from instances in your VPC to the internet** **(Correct)**

○ **Configure a Carrier gateway, that allows outbound communication over IPv6 from instances in your VPC to the internet**

○ **Configure an Internet Gateway to allow outbound communication on IPv6. Associate the IPv6 address to an Elastic IP address to make it public**

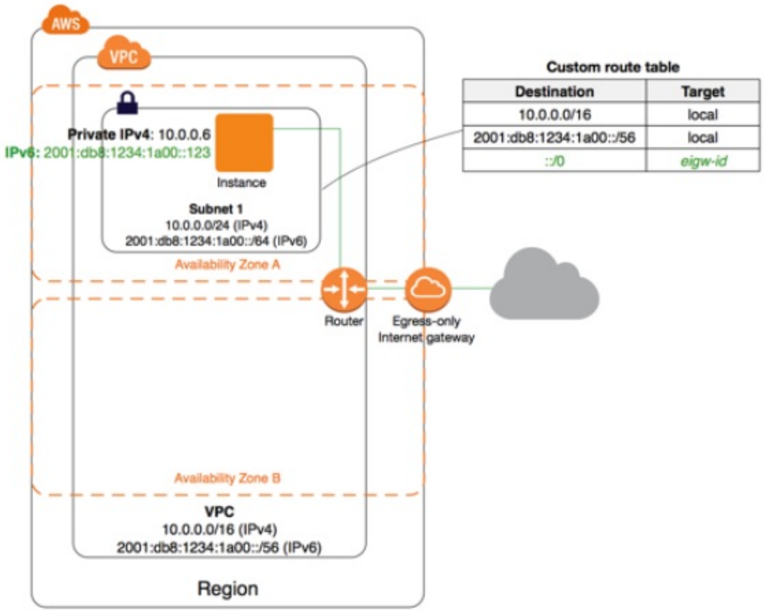
Explanation

Correct option:

Configure Egress-only Internet Gateway, that allows outbound communication over IPv6 from instances in your VPC to the internet - An egress-only internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the internet, and prevents the internet from initiating an IPv6 connection with your instances.

An egress-only internet gateway is for use with IPv6 traffic only. To enable outbound-only internet communication over IPv4, use a NAT gateway instead.

An example Egress-only Internet Gateway architecture:



Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
::/0	*eigw-id*

via

- https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html

Incorrect options:

Configure a Carrier gateway, that allows outbound communication over IPv6 from instances in your VPC to the internet - Carrier gateways are only available for VPCs that contain subnets in a Wavelength Zone. The carrier gateway provides connectivity between your Wavelength Zone and the telecommunication carrier, and devices on the telecommunication carrier network. A carrier gateway supports only IPv4 traffic.

Configure an Internet Gateway to allow outbound communication on IPv6. Associate the IPv6 address to an Elastic IP address to make it public - IPv6 addresses are globally unique and are therefore public by default. You do not associate them with an Elastic IP address.

Configure an Internet Gateway to allow outbound communication on IPv6 for the instances in the private subnet for your VPC. Public subnets are by default connected to the internet and do not need any extra configuration - Internet Gateway is configured to public subnets of a VPC and not to a private subnet directly.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html

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

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

Question 15: **Incorrect**

As part of the ongoing system maintenance, a SysOps Administrator has decided to increase the storage capacity of an EBS volume that is attached to an Amazon EC2 instance. However, the increased size is not reflected in the file system.

What has gone wrong in the configuration and how can it be fixed?

○ **EBS volume needs to be detached and attached back again to the instance for the modifications to show**

○ **Linux servers automatically pick the modifications done to EBS volumes, but Windows servers do not offer this feature. Use the Windows Disk Management utility to increase the disk size to the new modified volume size** **(Incorrect)**

○ **After you increase the size of an EBS volume, you must extend the file system to a larger size** (Correct)

○ **EBS volume might be encrypted. Encrypted EBS volumes will not show modifications done when still attached to the instance. Detach the EBS volume and attach it back**

Explanation

Correct option:

After you increase the size of an EBS volume, you must extend the file system to a larger size - After you increase the size of an EBS volume, you must use the file-system specific commands to extend the file system to the larger size. You can resize the file system as soon as the volume enters the optimizing state.

The process for extending a file system on Linux is as follows:

1. Your EBS volume might have a partition that contains the file system and data. Increasing the size of a volume does not increase the size of the partition. Before you extend the file system on a resized volume, check whether the volume has a partition that must be extended to the new size of the volume.
2. Use a file system-specific command to resize each file system to the new volume capacity.

Incorrect options:

EBS volume needs to be detached and attached back again to the instance for the modifications to show - This is incorrect and has been added as a distractor.

EBS volume might be encrypted. Encrypted EBS volumes will not show modifications done when still attached to the instance. Detach the EBS volume and attach it back - EBS volume encryption has no bearing on the given scenario.

Linux servers automatically pick the modifications done to EBS volumes, but Windows servers do not offer this feature. Use the Windows Disk Management utility to increase the disk size to the new modified volume size - As discussed above, You need to manually extend the size of the file system after increasing the size of EBS volume.

On the Windows file system, after you increase the size of an EBS volume, use the Windows Disk Management utility or PowerShell to extend the disk size to the new size of the volume. You can begin resizing the file system as soon as the volume enters the optimizing state.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html

https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/recognize-expanded-volume-windows.html

Question 16: **Incorrect**

A financial services company runs its server infrastructure on a fleet of Amazon EC2 instances running behind an Auto Scaling Group (ASG). The SysOps Administrator has configured the instances to be protected from termination during scale-in.

A scale-in event has occurred. What is the outcome of the event?

○ **The minimum capacity of the ASG is decremented, but ASG will not be able to terminate any instance** (Incorrect)

○ **When all instances are termination protected, scale-in event is not generated**

○ **The desired capacity of the ASG is decremented and the instances are terminated based on the configuration**

○ **The desired capacity of the ASG is decremented, but ASG will not be able to terminate any instance** **(Correct)**

Explanation

Correct option:

The desired capacity of the ASG is decremented, but ASG will not be able to terminate any instance - To control whether an Auto Scaling group can terminate a particular instance when scaling in, use instance scale-in protection. You can enable the instance scale-in protection setting on an Auto Scaling group or an individual Auto Scaling instance.

If all instances in an Auto Scaling group are protected from termination during scale in, and a scale-in event occurs, its desired capacity is decremented. However, the Auto Scaling group can't terminate the required number of instances until their instance scale-in protection settings are disabled.

Instance scale-in protection does not protect Auto Scaling instances from the following:

1. Manual termination through the Amazon EC2 console, the `terminate-instances` command, or the TerminateInstances action. To protect Auto Scaling instances from manual termination, enable Amazon EC2 termination protection.
2. Health check replacement if the instance fails health checks. To prevent Amazon EC2 Auto Scaling from terminating unhealthy instances, suspend the ReplaceUnhealthy process.
3. Spot Instance interruptions. A Spot Instance is terminated when capacity is no longer available or the Spot price exceeds your maximum price.

Incorrect options:

The minimum capacity of the ASG is decremented, but ASG will not be able to terminate any instance

The desired capacity of the ASG is decremented and the instances are terminated based on the configuration

When all instances are termination protected, scale-in event is not generated

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

Reference:

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

Question 17: **Correct**

A large IT company manages several projects on AWS Cloud and has decided to use AWS X-Ray to trace application workflows. The company uses a plethora of AWS services like API Gateway, Amazon EC2 instances, Amazon S3 storage service, Elastic Load Balancers and AWS Lambda functions.

Which of the following should the company keep in mind while using AWS X-Ray for the AWS services they use?

○ **You cannot use X-Ray to trace or analyze user requests to your Amazon API Gateway APIs**

○ **AWS X-Ray does not integrate with Amazon S3 and you need to use CloudTrail for tracking requests on S3**

○ **Application Load balancers do not send data to X-Ray** **(Correct)**

○ **AWS X-Ray cannot be used to trace your AWS Lambda functions since they are not integrated**

Explanation

Correct option:

Application Load balancers do not send data to X-Ray - Elastic Load Balancing application load balancers add a trace ID to incoming HTTP requests in a header named X-Amzn-Trace-Id. Load balancers do not send data to X-Ray and do not appear as a node on your service map.

Incorrect options:

AWS X-Ray does not integrate with Amazon S3 and you need to use CloudTrail for tracking requests on S3 - AWS X-Ray integrates with Amazon S3 to trace upstream requests to update your application's S3 buckets.

AWS X-Ray cannot be used to trace your AWS Lambda functions since they are not integrated - You can use AWS X-Ray to trace your AWS Lambda functions. Lambda runs the X-Ray daemon and records a segment with details about the function invocation and execution.

You cannot use X-Ray to trace or analyze user requests to your Amazon API Gateway APIs - You can use X-Ray to trace and analyze user requests as they travel through your Amazon API Gateway APIs to the underlying services. API Gateway supports X-Ray tracing for all API Gateway endpoint types: Regional, edge-optimized, and private. You can use X-Ray with Amazon API Gateway in all AWS Regions where X-Ray is available.

Reference:

https://docs.aws.amazon.com/xray/latest/devguide/xray-services-elb.html

Question 18: **Correct**

The development team at a retail company manages the deployment and scaling of their web application through AWS Elastic Beanstalk. After configuring the Elastic Beanstalk environment, the team has realized that Beanstalk is not handling the scaling activities the way they expected. This has impacted the application's ability to respond to the variations in traffic.

How should the environment be configured to get the best of Beanstalk's auto-scaling capabilities?

○ **The Auto Scaling group in your Elastic Beanstalk environment uses two default Amazon CloudWatch alarms to trigger scaling operations. These alarms must be configured based on the parameters appropriate for your application** **(Correct)**

○ **By default, Auto Scaling group created from Beanstalk uses Elastic Load Balancing health checks. Configure the Beanstalk to use Amazon EC2 status checks**

○ **The IAM Role attached to the Auto Scaling group might not have enough permissions to scale instances on-demand**

○ **The Auto Scaling group in your Elastic Beanstalk environment uses the number of logged-in users, as the criteria to trigger auto-scaling action. These alarms must be configured based on the parameters appropriate for your application**

Explanation

Correct option:

The Auto Scaling group in your Elastic Beanstalk environment uses two default Amazon CloudWatch alarms to trigger scaling operations. These alarms must be configured based on the parameters appropriate for your application

The Auto Scaling group in your Elastic Beanstalk environment uses two Amazon CloudWatch alarms to trigger scaling operations. Default Auto Scaling triggers are configured to scale when the average outbound network traffic (NetworkOut) from each instance is higher than 6 MB or lower than 2 MB over a period of five minutes.

For more efficient Amazon EC2 Auto Scaling, configure triggers that are appropriate for your application, instance type, and service requirements. You can scale based on several statistics including latency, disk I/O, CPU utilization, and request count.

Incorrect options:

The IAM Role attached to the Auto Scaling group might not have enough permissions to scale instances on-demand - The Auto Scaling group will not be able to spin up Amazon EC2 instances if the IAM Role associated with Beanstalk does not have enough permissions. Since the current use-case talks about scaling not happening at the expected rate, this should not be the issue.

By default, Auto Scaling group created from Beanstalk uses Elastic Load Balancing health checks. Configure the Beanstalk to use Amazon EC2 status checks - This statement is incorrect. By default, Auto Scaling group created from Beanstalk uses Amazon EC2 status checks.

The Auto Scaling group in your Elastic Beanstalk environment uses the number of logged-in users, as the criteria to trigger auto-scaling action. These alarms must be configured based on the parameters appropriate for your application - The default scaling criteria has already been discussed above (and it is not the number of logged-in users).

Reference:

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

Question 19: **Correct**

After configuring Amazon EC2 Auto Scaling, a systems administrator had tried to launch the Auto Scaling Group. But, the following launch failure message was displayed - `Client.InternalError: Client error on launch`.

What is the cause of this error and how can it be fixed?

○ **The security group specified in your launch configuration might have been deleted**

○ **Your cluster placement group contains an invalid instance type**

○ **The block device mappings in your launch configuration might contain block device names that are not available or currently not supported**

○ **This error can be caused when an Auto Scaling group attempts to launch an instance that has an encrypted EBS** **(Correct)**

volume, but the service-linked role does not have access to the customer-managed CMK used to encrypt it

Explanation

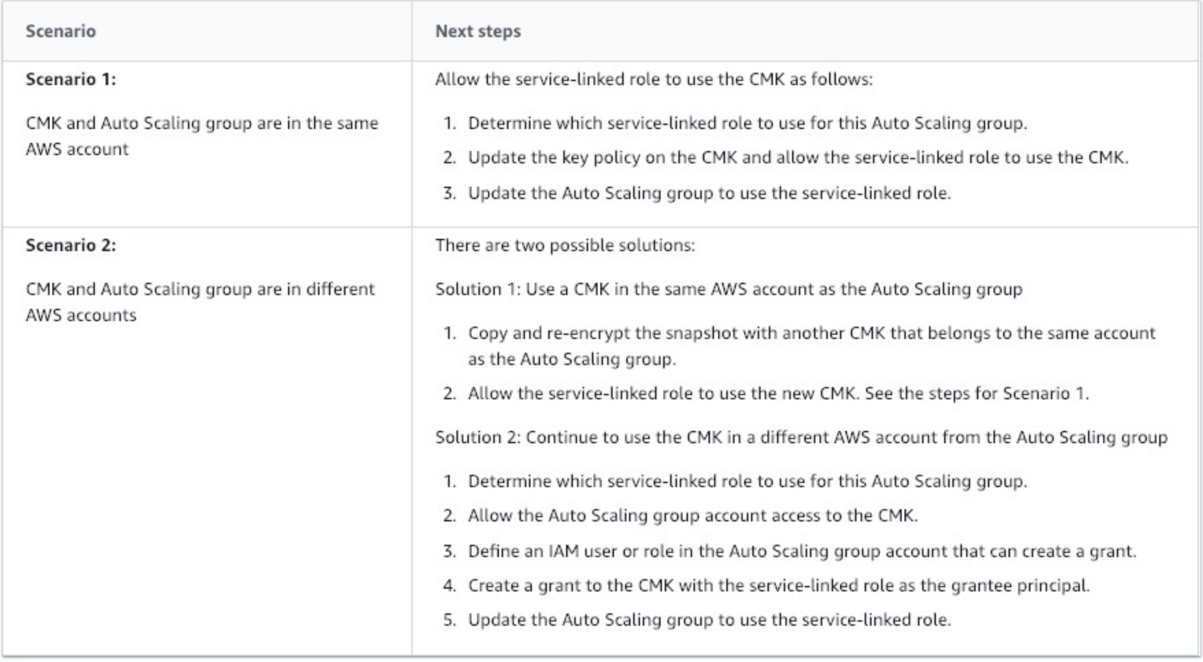
Correct option:

This error can be caused when an Auto Scaling group attempts to launch an instance that has an encrypted EBS volume, but the service-linked role does not have access to the customer-managed CMK used to encrypt it

`Client.InternalError: Client error on launch` error is caused when an Auto Scaling group attempts to launch an instance that has an encrypted EBS volume, but the service-linked role does not have access to the customer-managed CMK used to encrypt it. Additional setup is required to allow the Auto Scaling group to launch instances.

There are two scenarios possible: 1)CMK and Auto Scaling group are in the same AWS account, 2)CMK and Auto Scaling group are in different AWS accounts.

Full instructions for configuring the above two scenarios:



Scenario	Next steps
Scenario 1: CMK and Auto Scaling group are in the same AWS account	Allow the service-linked role to use the CMK as follows: 1. Determine which service-linked role to use for this Auto Scaling group. 2. Update the key policy on the CMK and allow the service-linked role to use the CMK. 3. Update the Auto Scaling group to use the service-linked role.
Scenario 2: CMK and Auto Scaling group are in different AWS accounts	There are two possible solutions: Solution 1: Use a CMK in the same AWS account as the Auto Scaling group 1. Copy and re-encrypt the snapshot with another CMK that belongs to the same account as the Auto Scaling group. 2. Allow the service-linked role to use the new CMK. See the steps for Scenario 1. Solution 2: Continue to use the CMK in a different AWS account from the Auto Scaling group 1. Determine which service-linked role to use for this Auto Scaling group. 2. Allow the Auto Scaling group account access to the CMK. 3. Define an IAM user or role in the Auto Scaling group account that can create a grant. 4. Create a grant to the CMK with the service-linked role as the grantee principal. 5. Update the Auto Scaling group to use the service-linked role.

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/ts-as-instancelaunchfailure.html#ts-as-instancelaunchfailure-12

Incorrect options:

The security group specified in your launch configuration might have been deleted - This configuration will generate an error like so - "The security group <name of the security group> does not exist. Launching EC2 instance failed."

The block device mappings in your launch configuration might contain block device names that are not available or currently not supported - This configuration will generate an error like so - "Invalid device name upload. Launching EC2 instance failed."

Your cluster placement group contains an invalid instance type - This configuration will generate an error like so - "Placement groups may not be used with instances of type 'm1.large'. Launching EC2 instance failed."

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/ts-as-instancelaunchfailure.html#ts-as-instancelaunchfailure-12

Question 20: **Correct**

As a SysOps Administrator, you have been asked to calculate the total network usage for all the EC2 instances of a company and determine which instance used the most bandwidth within a date range.

Which Amazon CloudWatch metric(s) will help you get the needed data?

○ `DiskReadBytes` **and** `DiskWriteBytes`

○ `NetworkIn` and `NetworkOut` **(Correct)**

○ `DataTransfer-Out-Bytes`

○ `NetworkTotalBytes`

Explanation

Correct option:

`NetworkIn` and `NetworkOut` - You can determine which instance is causing high network usage using the Amazon CloudWatch NetworkIn and NetworkOut metrics. You can aggregate the data points from these metrics to calculate the network usage for your instance.

`NetworkIn` - The number of bytes received by the instance on all network interfaces. This metric identifies the volume of incoming network traffic to a single instance.

The number reported is the number of bytes received during the period. If you are using basic (five-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring and the statistic is Sum, divide it by 60. Units of this metric are Bytes.

`NetworkOut` - The number of bytes sent out by the instance on all network interfaces. This metric identifies the volume of outgoing network traffic from a single instance.

The number reported is the number of bytes sent during the period. If you are using basic (five-minute) monitoring and the statistic is Sum, you can divide this number by 300 to find Bytes/second. If you have detailed (one-minute) monitoring and the statistic is Sum, divide it by 60. Units of this metric are Bytes.

Incorrect options:

`DataTransfer-Out-Bytes` - `DataTransfer-Out-Bytes` metric is used in AWS Cost Explorer reports and is not useful for the current scenario.

`DiskReadBytes` and `DiskWriteBytes` - `DiskReadBytes` is the bytes read from all instance store volumes available to the instance. This metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application.

`DiskWriteBytes` is the bytes written to all instance store volumes available to the instance. This metric is used to determine the volume of the data the application writes onto the hard disk of the instance. This can be used to determine the speed of the application.

`NetworkTotalBytes` - This is a made-up option, given only as a distractor.

Reference:

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

Question 21: **Correct**

A systems administrator has configured Amazon EC2 instances in an Auto Scaling Group (ASG) for two separate development teams. However, only one configuration has CloudWatch agent installed on the instances, whereas the other one does not have it. The administrator has not manually installed the agents on either group of instances.

Which of the following would you identify as a root-cause behind this issue?

○ **The instance architecture might not have been compatible with the AMI chosen. The incompatibility results in various errors, one of which is, some of the AWS services will not be installed as expected**

○ **CloudWatch agent can be configured to be loaded on the EC2 instances while configuring the ASG. The developer could have unintentionally checked this flag on one of the ASGs he created**

○ **If your AMI contains a CloudWatch agent, it's automatically installed on EC2 instances when you create an EC2 Auto Scaling group. The developer needs to choose the AMI that has CloudWatch agent pre-configured on it** **(Correct)**

○ **The architecture of the `InstanceType` mentioned in your launch configuration does not match the image architecture. So, the ASG was created with errors, resulting in skipping CloudWatch agent. A thorough check is needed for such ASGs, more services could have been skipped**

Explanation

Correct option:

If your AMI contains a CloudWatch agent, it's automatically installed on EC2 instances when you create an EC2 Auto Scaling group. The developer needs to choose the AMI that has CloudWatch agent pre-configured on it

If your AMI contains a CloudWatch agent, it's automatically installed on EC2 instances when you create an EC2 Auto Scaling group. With the stock Amazon Linux AMI, you need to install it (AWS recommends to install via yum).

Incorrect options:

CloudWatch agent can be configured to be loaded on the EC2 instances while configuring the ASG. The developer could have unintentionally checked this flag on one of the ASGs he created - This is incorrect and added only as a distractor.

The architecture of the `InstanceType` mentioned in your launch configuration does not match the image architecture. So, the ASG was created with errors, resulting in skipping CloudWatch agent. A thorough check is needed for such ASGs, more services could have been skipped - This is incorrect. Either the ASG is created successfully or fails completely. Partial installation of services will not take place.

The instance architecture might not have been compatible with the AMI chosen. The incompatibility results in various errors, one of which is, some of the AWS services will not be installed as expected - If there are compatibility issues, the ASG will not be able to spin up instances, and throws an error that explains the compatibility error.

Reference:

https://aws.amazon.com/ec2/autoscaling/faqs/

Question 22: **Incorrect**

A highly critical financial services application is being moved to AWS Cloud from the on-premises data center. The application uses a fleet of Amazon EC2 instances provisioned in different geographical areas. The Chief Technology Officer (CTO) of the company needs to understand the communication network used between instances at various locations when they interact using public IP addresses.

Which of the following options would you identify as correct? (Select two)

Direct Connect is the default way of communication where there is no Inter-Region VPC Peering connection between the VPCs. All traffic between instances will use Direct Connect and does not go over the internet (Incorrect)

Traffic between EC2 instances in different AWS Regions where there is no Inter-Region VPC Peering connection between the VPCs where these instances reside, will use edge locations to communicate without going over the internet

Traffic between two EC2 instances always stays within the AWS network, even when it goes over public IP addresses by using AWS Global Infrastructure

Traffic between EC2 instances in different AWS Regions stays within the AWS network, if there is an Inter-Region VPC Peering connection between the VPCs where the two instances reside (Correct)

Traffic between two EC2 instances in the same AWS Region stays within the AWS network, even when it goes over public IP addresses (Correct)

Explanation

Correct option:

Traffic between EC2 instances in different AWS Regions stays within the AWS network, if there is an Inter-Region VPC Peering connection between the VPCs where the two instances reside

Traffic between two EC2 instances in the same AWS Region stays within the AWS network, even when it goes over public IP addresses

When two instances communicate using public IP addresses, the following three scenarios are possible: 1. Traffic between two EC2 instances in the same AWS Region stays within the AWS network, even when it goes over public IP addresses.

1. Traffic between EC2 instances in different AWS Regions stays within the AWS network if there is an Inter-Region VPC Peering connection between the VPCs where the two instances reside.

2. Traffic between EC2 instances in different AWS Regions where there is no Inter-Region VPC Peering connection between the VPCs where these instances reside, is not guaranteed to stay within the AWS network.

Incorrect options:

Traffic between two EC2 instances always stays within the AWS network, even when it goes over public IP addresses by using AWS Global Infrastructure

Traffic between EC2 instances in different AWS Regions where there is no Inter-Region VPC Peering connection between the VPCs where these instances reside will use edge locations to communicate without going over the internet

These two options contradict the explanation provided above, so both options are incorrect.

Direct Connect is the default way of communication where there is no Inter-Region VPC Peering connection between the VPCs. All traffic between instances will use Direct Connect and does not go over the internet - AWS Direct Connect is a network service that provides an alternative to using the Internet to utilize AWS cloud services. AWS Direct Connect enables customers to have low latency and private connections to AWS for workloads that require higher speed or lower latency than the internet. Direct Connect is a paid service and is available only if the customer opts for it.

Reference:

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

Question 23: **Incorrect**

An automobile company uses a hybrid environment to run its technology infrastructure using a mix of on-premises instances and AWS Cloud. The company has a few managed instances in Amazon VPC. The company wants to avoid using the internet for accessing AWS Systems Manager APIs from this VPC.

As a Systems Administrator, which of the following would you recommend to address this requirement?

○ **You can privately access AWS Systems Manager APIs from Amazon VPC by creating VPC Endpoint** **(Correct)**

○ **You can privately access AWS Systems Manager APIs from Amazon VPC by creating Internet Gateway**

○ **You can privately access AWS Systems Manager APIs from Amazon VPC by creating VPN connection** **(Incorrect)**

○ **You can privately access AWS Systems Manager APIs from Amazon VPC by creating NAT gateway**

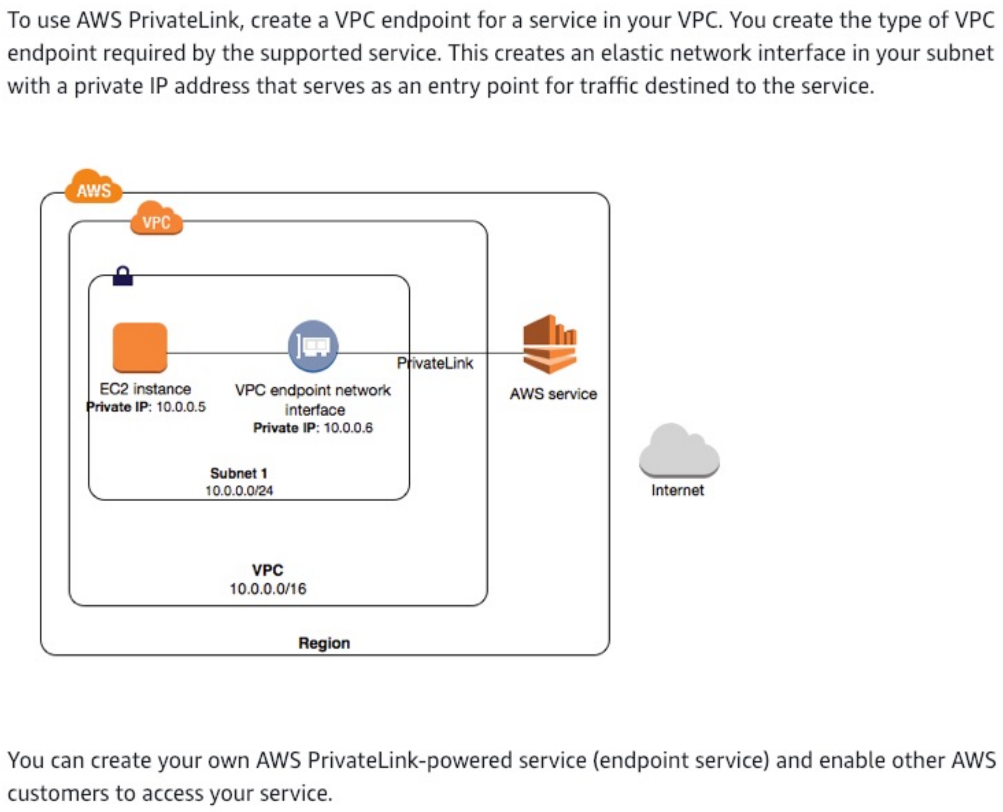
Explanation

Correct option:

You can privately access AWS Systems Manager APIs from Amazon VPC by creating VPC Endpoint - A managed instance is any machine configured for AWS Systems Manager. You can configure EC2 instances or on-premises machines in a hybrid environment as managed instances.

You can improve the security posture of your managed instances (including managed instances in your hybrid environment) by configuring AWS Systems Manager to use an interface VPC endpoint in Amazon Virtual Private Cloud (Amazon VPC). An interface VPC endpoint (interface endpoint) enables you to connect to services powered by AWS PrivateLink, a technology that enables you to privately access Amazon EC2 and Systems Manager APIs by using private IP addresses. PrivateLink restricts all network traffic between your managed instances, Systems Manager, and Amazon EC2 to the Amazon network. This means that your managed instances don't have access to the Internet. If you use PrivateLink, you don't need an Internet gateway, a NAT device, or a virtual private gateway.

How to use AWS PrivateLink:



To use AWS PrivateLink, create a VPC endpoint for a service in your VPC. You create the type of VPC endpoint required by the supported service. This creates an elastic network interface in your subnet with a private IP address that serves as an entry point for traffic destined to the service.

You can create your own AWS PrivateLink-powered service (endpoint service) and enable other AWS customers to access your service.

via

- https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html#what-is-privatelink

Incorrect options:

You can privately access AWS Systems Manager APIs from Amazon VPC by creating 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 to the route table.

You can privately access AWS Systems Manager APIs from Amazon VPC by creating 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.

You can privately access AWS Systems Manager APIs from Amazon VPC by creating VPN connection

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.

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

Reference:

https://docs.aws.amazon.com/systems-manager/latest/userguide/setup-create-vpc.html

Question 24: **Incorrect**

As a SysOps Administrator, you create and maintain various system configurations for the teams you work with. You have created a CloudFront distribution with origin as an Amazon S3 bucket. The configuration has worked fine so far. However, for a few hours now, an error similar to this has cropped up - `The authorization header is malformed; the region '<AWS Region>' is wrong; expecting '<AWS Region>'`.

What is the reason for this error and how will you fix it?

○ **This error indicates that the CloudFront distribution and Amazon S3 are not in the same AWS Region. Move one resource so that, both the CloudFront distribution and Amazon S3 are in the same AWS Region**

(Incorrect)

○ **This error indicates that the API key used for authorization is from an AWS Region that is different from the Region that S3 bucket is created in**

○ **This error indicates the configured Amazon S3 bucket has been moved from one AWS Region to the other. That is, deleted from one AWS Region and created with the same name in another. To fix this error, update your CloudFront distribution so that it finds the S3 bucket in the bucket's current AWS Region** **(Correct)**

○ **This error indicates that when CloudFront forwarded a request to the origin, the origin didn’t respond before the request expired. This could be an access issue caused by a firewall or a Security Group not allowing access to CloudFront to access S3 resources**

Explanation

Correct option:

This error indicates the configured Amazon S3 bucket has been moved from one AWS Region to the other. That is, deleted from one AWS Region and created with the same name in another. To fix this error, update your CloudFront distribution so that it finds the S3 bucket in the bucket's current AWS Region - If CloudFront requests an object from your origin, and the origin returns an HTTP 4xx or 5xx status code, there's a problem with communication between CloudFront and your origin.

Your CloudFront distribution might send error responses with HTTP status code 400 Bad Request, and a message similar to the following: `The authorization header is malformed; the region '<AWS Region>' is wrong; expecting '<AWS Region>'`.

This problem can occur in the following scenario: 1)Your CloudFront distribution's origin is an Amazon S3 bucket, 2)You moved the S3 bucket from one AWS Region to another. That is, you deleted the S3 bucket, then later you created a new bucket with the same bucket name, but in a different AWS Region than where the original S3 bucket was located.

To fix this error, update your CloudFront distribution so that it finds the S3 bucket in the bucket's current AWS Region.

Incorrect options:

This error indicates that the CloudFront distribution and Amazon S3 are not in the same AWS Region. Move one resource so that, both the CloudFront distribution and Amazon S3 are in the same AWS Region - Amazon CloudFront uses a global network of edge locations and regional edge caches for content delivery. You can configure CloudFront to server content from particular Regions, but CloudFront is not Region-specific.

This error indicates that the API key used for authorization is from an AWS Region that is different from the Region that S3 bucket is created in - This is a made-up option, given only as a distractor.

This error indicates that when CloudFront forwarded a request to the origin, the origin didn’t respond before the request expired. This could be an access issue caused by a firewall or a Security Group not allowing access to CloudFront to access S3 resources - When CloudFront forwards a request to the origin, and the origin didn’t respond before the request expired, a Gateway Timeout error is generated.

Reference:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/http-400-bad-request.html

Question 25: **Incorrect**

As part of regular maintenance, a systems administrator was checking through the configured Auto Scaling groups (ASGs). An error was raised by an Auto Scaling group when attempting to launch an instance that has an encrypted EBS volume. The service-linked role did not have access to the customer-managed CMK used to encrypt the volume.

Which of the following represents the best solution to fix this issue?

○ **Determine which service-linked role to use for this Auto Scaling group. Update the key policy on the CMK and allow the service-linked role to use the CMK. Update the Auto Scaling group to use the service-linked role** **(Incorrect)**

○ **Export the CMK to the ASG account from the instance account. Then, define a role to access this CMK and attach the role to ASG**

○ **It is not possible for ASGs to initiate EC2 instances that have encrypted volumes attached to them**

○ **Use a CMK in the same AWS account as the Auto Scaling group (ASG). Copy and re-encrypt the snapshot with another CMK that belongs to the same account as the Auto Scaling group. Allow the service-linked role to use the new CMK** **(Correct)**

Explanation

Correct option:

Use a CMK in the same AWS account as the Auto Scaling group (ASG). Copy and re-encrypt the snapshot with another CMK that belongs to the same account as the Auto Scaling group. Allow the service-linked role to use the new CMK

`Client.InternalError: Client error on launch` error is thrown when an Auto Scaling group attempts to launch an instance that has an encrypted EBS volume, but the service-linked role does not have access to the customer-managed CMK used to encrypt it.

There are two possible solutions:

Solution 1: Use a CMK in the same AWS account as the Auto Scaling group. Copy and re-encrypt the snapshot with another CMK that belongs to the same account as the Auto Scaling group. Allow the service-linked role to use the new CMK.

Solution 2: Continue to use the CMK in a different AWS account from the Auto Scaling group. Determine which service-linked role to use for this Auto Scaling group. Allow the Auto Scaling group account access to the CMK. Define an IAM user or role in the Auto Scaling group account that can create a grant. Create a grant to the CMK with the service-linked role as the grantee principal. Update the Auto Scaling group to use the service-linked role.

Incorrect options:

Determine which service-linked role to use for this Auto Scaling group. Update the key policy on the CMK and allow the service-linked role to use the CMK. Update the Auto Scaling group to use the service-linked role - This is possible only when CMK and Auto Scaling group are in the same AWS account.

Export the CMK to the ASG account from the instance account. Then, define a role to access this CMK and attach the role to ASG - It is not possible to export CMKs.

It is not possible for ASGs to initiate EC2 instances that have encrypted volumes attached to them - This statement is incorrect and only given as a distractor.

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/ts-as-instancelaunchfailure.html#ts-as-instancelaunchfailure-10

Question 26: **Correct**

A large IT project has multiple teams working on it. The teams share access across the resources - Amazon EC2 instances, Amazon S3 buckets and RDS database. A junior developer of a team ended up deleting data from a bucket that was used by various teams. This resulted in significant wastage of time and resources to mitigate the situation.

As a SysOps Administrator, you have been hired to secure the data present in S3 buckets by allowing recovery of objects in case of an accidental deletion. Which of these options would you suggest to meet the given requirements?

○ **Enable S3 Object Lock to lock all the objects that are used in the project**

○ **Use Amazon S3 replication to replicate critical objects to avoid losing them from unintended deletes**

○ **Use Amazon S3 bucket owner condition to restrict access to unintended users**

○ **Enable S3 versioning on all the buckets used in the project** **(Correct)**

Explanation

Correct option:

Enable S3 versioning on all the buckets used in the project - 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.

Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. For example:

1. If you delete an object, instead of removing it permanently, Amazon S3 inserts a delete marker, which becomes the current object version. You can always restore the previous version.
2. If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

Incorrect options:

Enable S3 Object Lock to lock all the objects that are used in the project - With S3 Object Lock, you can store objects using a write-once-read-many (WORM) model. You can use it to prevent an object from being deleted or overwritten for a fixed amount of time or indefinitely. This is not ideal for the given requirement where developers from different teams need to change the objects in the S3 buckets.

Use Amazon S3 bucket owner condition to restrict access to unintended users - Amazon S3 bucket owner condition ensures that the buckets you use in your S3 operations belong to the AWS accounts that you expect. Bucket owner condition enables you to verify that the target bucket is owned by the expected AWS account. As much as this is a useful feature, it's not helpful for the current scenario.

Use Amazon S3 replication to replicate critical objects to avoid losing them from unintended deletes - Replication enables automatic, asynchronous copying of objects across Amazon S3 buckets. Replication is generally used to safeguard from failure or copy data across different environments (like production, testing, development). This is not helpful for the given scenario.

References:

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

https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock.html

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

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

Question 27: **Correct**

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

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

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

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

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

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

Explanation

Correct option:

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

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

Incorrect options:

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

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

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

Reference:

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

Question 28: **Incorrect**

A retail company has built its server infrastructure on Amazon EC2 instances that run on Windows OS. The development team has defined a few custom metrics that need to be collected by the unified CloudWatch agent.

As a SysOps Administrator, can you identify the correct configuration to be used for this scenario?

○ **CloudWatch agent can be configured with either StatsD protocol or collectd protocol to collect the necessary system metrics on windows servers**

○ **Configure the CloudWatch agent with StatsD protocol to collect the necessary system metrics** **(Correct)**

○ **Configure the CloudWatch agent with collectd protocol to collect the necessary system metrics** **(Incorrect)**

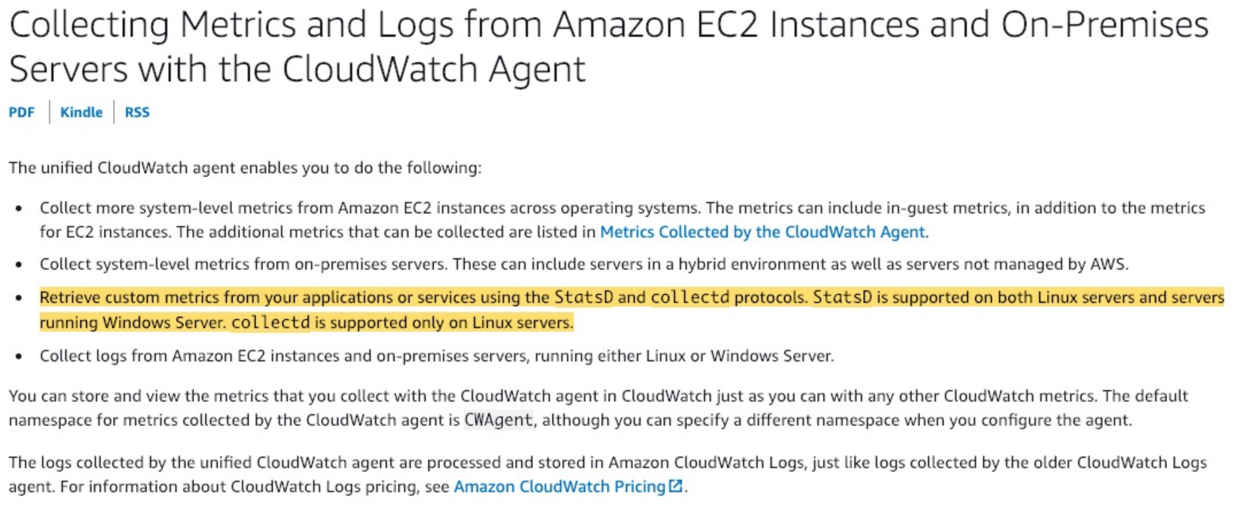
○ **Unified CloudWatch agent cannot be custom configured**

Explanation

Correct option:

Configure the CloudWatch agent with StatsD protocol to collect the necessary system metrics - You can retrieve custom metrics from your applications or services using the StatsD and collectd protocols. StatsD is supported on both Linux servers and servers running Windows Server. collectd is supported only on Linux servers. Here, the instances are running on Windows servers, hence StatsD is the right protocol.

More information on Collecting Metrics and Logs from Amazon EC2 Instances:



Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent

PDF | Kindle | RSS

The unified CloudWatch agent enables you to do the following:

- Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances. The additional metrics that can be collected are listed in Metrics Collected by the CloudWatch Agent.
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS.
- Retrieve custom metrics from your applications or services using the `StatsD` and `collectd` protocols. `StatsD` is supported on both Linux servers and servers running Windows Server. `collectd` is supported only on Linux servers.
- Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server.

You can store and view the metrics that you collect with the CloudWatch agent in CloudWatch just as you can with any other CloudWatch metrics. The default namespace for metrics collected by the CloudWatch agent is `CWAgent`, although you can specify a different namespace when you configure the agent.

The logs collected by the unified CloudWatch agent are processed and stored in Amazon CloudWatch Logs, just like logs collected by the older CloudWatch Logs agent. For information about CloudWatch Logs pricing, see Amazon CloudWatch Pricing.

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

Incorrect options:

Configure the CloudWatch agent with collectd protocol to collect the necessary system metrics - collectd is supported only on Linux servers and hence it is not the correct choice here.

CloudWatch agent can be configured with either StatsD protocol or collectd protocol to collect the necessary system metrics on windows servers - StatsD is supported on both Linux servers and servers running Windows Server. collectd is supported only on Linux servers.

Unified CloudWatch agent cannot be custom configured - This is an incorrect statement and used only as a distractor.

Reference:

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

Question 29: **Incorrect**

A developer has created rules for different events on Amazon EventBridge with AWS Lambda function as a target. The developer has also created an IAM Role with the necessary permissions and associated it with the rule. The rule however is failing, and on initial analysis, it is clear that the IAM Role associated with the rule is not being used when calling the Lambda function.

What could have gone wrong with the configuration and how can you fix the issue?

○ **For Lambda functions configured as a target to EventBridge, you need to provide resource-based policy. IAM Roles will not work** **(Correct)**

○ **The IAM Role is wrongly configured. Delete the existing Role and recreate with necessary permissions and associate the newly created Role with the EventBridge rule**

○ **For Lambda, EventBridge relies on Access Control Lists (ACLs) to define permissions. IAM Roles will not work for Lambda when configured as a target for an EventBridge rule**

○ **AWS Command Line Interface (CLI) should not be used to add permissions to EventBridge targets** **(Incorrect)**

Explanation

Correct option:

For Lambda functions configured as a target to EventBridge, you need to provide resource-based policy. IAM Roles will not work - IAM roles for rules are only used for events related to Kinesis Streams. For Lambda functions and Amazon SNS topics, you need to provide resource-based permissions.

When a rule is triggered in EventBridge, all the targets associated with the rule are invoked. Invocation means invoking the AWS Lambda functions, publishing to the Amazon SNS topics, and relaying the event to the Kinesis streams. In order to be able to make API calls against the resources you own, EventBridge needs the appropriate permissions. For Lambda, Amazon SNS, Amazon SQS, and Amazon CloudWatch Logs resources, EventBridge relies on resource-based policies. For Kinesis streams, EventBridge relies on IAM roles.

Incorrect options:

The IAM Role is wrongly configured. Delete the existing Role and recreate with necessary permissions and associate the newly created Role with the EventBridge rule - This option has been added as a distractor.

For Lambda, EventBridge relies on Access Control Lists (ACLs) to define permissions. IAM Roles will not work for Lambda when configured as a target for an EventBridge rule - Access Control Lists are not used with EventBridge and ACLs are defined at the account level and not at the individual user level.

AWS Command Line Interface (CLI) should not be used to add permissions to EventBridge targets - This statement is incorrect. AWS CLI can be used to add permissions to targets for EventBridge rules.

References:

https://docs.aws.amazon.com/eventbridge/latest/userguide/resource-based-policies-eventbridge.html

https://docs.aws.amazon.com/eventbridge/latest/userguide/eventbridge-troubleshooting.html

Question 30: **Correct**

A production-ready application has just been deployed to Amazon EC2 instance that uses MySQL RDS as the database. The team is looking at making the RDS deployment highly available and failure-proof.

As a SysOps Administrator, can you suggest an easy and effective way of configuring this requirement?

○ **Scale up your DB instance when you are approaching storage capacity limits**

○ **Configure the RDS to be a multi Availability Zone (AZ) deployment** **(Correct)**

○ **Configure your JVM with a TTL value of no more than 60 seconds, to help you re-establish the connection to your database, in case of failure**

○ **Configure automated backups for the RDS instance, to retrieve data and instance status, if needed after a failure**

Explanation

Correct option:

Configure the RDS to be a multi Availability Zone (AZ) deployment - Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments.

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.

Using the RDS console, you can create a Multi-AZ deployment by simply specifying Multi-AZ when creating a DB instance. You can use the console to convert existing DB instances to Multi-AZ deployments by modifying the DB instance and specifying the Multi-AZ option. You can also specify a Multi-AZ deployment with the AWS CLI or Amazon RDS API. Use the create-db-instance or modify-db-instance CLI command, or the CreateDBInstance or ModifyDBInstance API operation.

Incorrect options:

Configure automated backups for the RDS instance, to retrieve data and instance status, if needed after a failure - The automated backup feature of Amazon RDS enables point-in-time recovery for your database instance. Amazon RDS will backup your database and transaction logs and store both for a user-specified retention period. Backups do not make the architecture highly available, a critical database should be deployed as a multi-AZ deployment, to cater to failures.

Scale up your DB instance when you are approaching storage capacity limits - This is vertical scaling and is not helpful when the requirement is high availability since there is still only one instance.

Configure your JVM with a TTL value of no more than 60 seconds, to help you re-establish the connection to your database, in case of failure - This change is part of high availability configuration and is needed when failover happens. But, multi -Z deployment is a pre-requisite for the DB architecture to be highly available.

Reference:

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

Question 31: **Correct**

An e-commerce company relies heavily on the AWS Systems Manager for automating various management tasks for the fleet of Amazon EC2 instances that host their applications.

Which of the following should you use to recover an impaired instance automatically?

○ **Use the `AWSSupport-ExecuteEC2Rescue` document to recover impaired instances** (Correct)

○ **Automatic recovery of impaired instances is not possible currently**

○ **Use the `AWS-UpdateCloudFormationStackWithApproval` document to update impaired instances**

○ **Use the `AWS-UpdateWindowsAmi` document to recover impaired instances**

Explanation

Correct option:

Use the `AWSSupport-ExecuteEC2Rescue` document to recover impaired instances

A Systems Manager Automation document defines the Automation workflow (the actions that Systems Manager performs on your managed instances and AWS resources). Automation includes several pre-defined Automation documents that you can use to perform common tasks like restarting one or more EC2 instances or creating an Amazon Machine Image (AMI).

Use the `AWSSupport-ExecuteEC2Rescue` document to recover impaired instances. An instance can become unreachable for a variety of reasons, including network misconfigurations, RDP issues, or firewall settings. Troubleshooting and regaining access to the instance previously required dozens of manual steps before you could regain access. The AWSSupport-ExecuteEC2Rescue document lets you regain access by specifying an instance ID and clicking a button.

Incorrect options:

Use the `AWS-UpdateCloudFormationStackWithApproval` document to update impaired instances - The `AWS-UpdateCloudFormationStackWithApproval` document is used to update resources that were deployed by using CloudFormation template.

Use the `AWS-UpdateWindowsAmi` document to recover impaired instances - You use the `AWS-UpdateLinuxAmi` and `AWS-UpdateWindowsAmi` documents to create golden AMIs from a source AMI.

Automatic recovery of impaired instances is not possible currently - This is an incorrect statement, added only as a distractor.

Reference:

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

Question 32: **Incorrect**

A retail company has realized that their Amazon EBS volume backed EC2 instance is consistently over-utilized and needs an upgrade. A developer has connected with you to understand the key parameters to be considered when changing the instance type.

As a SysOps Administrator, which of the following would you identify as correct regarding the instance types for the given use-case? (Select three)

Resizing of an instance is possible if the root device is either EBS volume or an instance store volume. However, instance store volumes taking longer to start on the new instance, since cache data is lost on these instances (Incorrect)

Resizing of an instance is only possible if the root device for your instance is an EBS volume (Correct)

There is no downtime on the instance if you choose an instance of a compatible type since AWS starts the new instance and shifts the applications from current instance (Incorrect)

You must stop your Amazon EBS–backed instance before you can change its instance type. AWS moves the instance to new hardware; however, the instance ID does not change (Correct)

If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance (Correct)

The new instance retains its public, private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses that were associated with the old instance

Explanation

Correct options:

Resizing of an instance is only possible if the root device for your instance is an EBS volume - If the root device for your instance is an EBS volume, you can change the size of the instance simply by changing its instance type, which is known as resizing it. If the root device for your instance is an instance store volume, you must migrate your application to a new instance with the instance type that you need.

You must stop your Amazon EBS–backed instance before you can change its instance type. AWS moves the instance to new hardware; however, the instance ID does not change - You must stop your Amazon EBS–backed instance before you can change its instance type. When you stop and start an instance, AWS moves the instance to new hardware; however, the instance ID does not change.

If your instance is in an Auto Scaling group, the Amazon EC2 Auto Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance - If your instance is in an Auto Scaling group, the Amazon EC2 Auto

Scaling service marks the stopped instance as unhealthy, and may terminate it and launch a replacement instance. To prevent this, you can suspend the scaling processes for the group while you're resizing your instance.

Incorrect options:

The new instance retains its public, private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses that were associated with the old instance - If your instance has a public IPv4 address, AWS releases the address and gives it a new public IPv4 address. The instance retains its private IPv4 addresses, any Elastic IP addresses, and any IPv6 addresses.

There is no downtime on the instance if you choose an instance of a compatible type since AWS starts the new instance and shifts the applications from current instance - AWS suggests that you plan for downtime while your instance is stopped. Stopping and resizing an instance may take a few minutes, and restarting your instance may take a variable amount of time depending on your application's startup scripts.

Resizing of an instance is possible if the root device is either EBS volume or an instance store volume. However, instance store volumes taking longer to start on the new instance, since cache data is lost on these instances - As discussed above, resizing of an instance is possible only if the root device for the instance is an EBS volume.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-resize.html

Question 33: **Incorrect**

A startup has reserved On-Demand Capacity Reservations for the Amazon EC2 instances they use for running analytics. Once the billing report was generated, the company was surprised to see that the costs were much higher than expected. The startup has hired you as a SysOps Administrator to bridge this knowledge gap.

Can you identify the important points to remember when considering On-Demand Capacity Reservations? (Select two)

□ **Capacity Reservations do not offer any billing discounts** **(Correct)**

□ **On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration** **(Correct)**

□ **On-Demand Capacity Reservations require a fixed one-year or three-year commitment**

□ **Capacity Reservations are transferable from one AWS account to another** **(Incorrect)**

Capacity Reservations can be used with Dedicated Hosts, however, they can't be used with placement groups

Explanation

Correct options:

On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration - On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or regional Reserved Instances.

Capacity Reservations do not offer any billing discounts - Capacity Reservations do not offer any billing discounts. You can combine Capacity Reservations with Savings Plans or Regional Reserved Instances to receive a discount.

Incorrect options:

On-Demand Capacity Reservations require a fixed one-year or three-year commitment - No commitment is required for On-Demand Capacity Reservations. They can be created and canceled as needed.

Capacity Reservations can be used with Dedicated Hosts, however, they can't be used with placement groups - Capacity Reservations can be used with neither placement groups nor Dedicated Hosts.

Capacity Reservations are transferable from one AWS account to another - Capacity Reservations are not transferable from one AWS account to another. However, you can share Capacity Reservations with other AWS accounts.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-capacity-reservations.html#capacity-reservations-differences

Question 34: **Correct**

AWS Shared Responsibility Model discusses the responsibilities that customers and AWS share for different services and resources.

For an abstracted service like Amazon S3, which of the following is the responsibility of AWS?

○ **Maintaining the operating systems and platforms for Amazon S3** **(Correct)**

○ **Choosing encryption options for data present in S3 buckets**

○ **Managing the data present in S3 Buckets**

○ **Defining rules to move data across different S3 storage classes**

Explanation

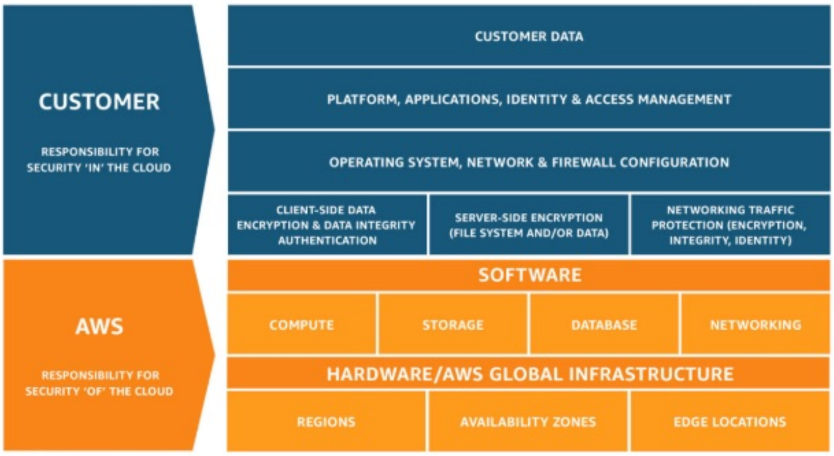
Correct option:

Maintaining the operating systems and platforms for Amazon S3 - Security and Compliance is a shared responsibility between AWS and the customer. AWS is responsible for “Security of the Cloud” and the customer is responsible for “Security in the Cloud”.

Customer responsibility will be determined by the AWS Cloud services that a customer selects. This determines the amount of configuration work that the customer must perform as part of their security responsibilities.

For abstracted services, such as Amazon S3 and Amazon DynamoDB, AWS operates the infrastructure layer, the operating system, and platforms, and customers access the endpoints to store and retrieve data. Customers are responsible for managing their data (including encryption options), classifying their assets, and using IAM tools to apply the appropriate permissions.

AWS Shared Responsibility Model: via - https://aws.amazon.com/compliance/shared-responsibility-model/


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

Incorrect options:

Managing the data present in S3 Buckets - This is the responsibility of the customer. Managing the storage infrastructure and redundantly storing data across AZs is the responsibility of AWS.

Choosing encryption options for data present in S3 buckets - Choosing an encryption option that suits the business requirements is the responsibility of the customer. Providing the different encryption options, maintaining and managing the security keys is the responsibility of AWS.

Defining rules to move data across different S3 storage classes - Defining rules to move data across storage classes is the responsibility of the customer. But, maintaining the backbone infrastructure of these storage classes - including hardware and software is the responsibility of AWS.

Reference:

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

Question 35: **Correct**

A small financial services startup uses Amazon EC2 instance for their server infrastructure and Amazon S3 for storage. The files stored on S3 are critical for the business and the company wants to track access to the buckets for audit and security purposes. The startup is looking at a cost-effective way of doing this without incurring extra costs.

As a Systems Administrator, which of the following would you recommend to address this use-case?

○ **Use Amazon Inspector, a security assessment service that helps track the calls made to AWS services configured with it**

○ **Enable Amazon S3 Server Access Logging for all the buckets that the company deems important and store these logs in another S3 bucket for analysis** **(Correct)**

○ **Use AWS CloudTrail to identify the requests made to the Amazon S3 buckets**

○ **Use AWS X-Ray for tracing Amazon S3 requests from end-to-end**

Explanation

Correct option:

Enable Amazon S3 Server Access Logging for all the buckets that the company deems important and store these logs in another S3 bucket for analysis - To track requests for access to your bucket, you can enable server access logging. 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.

There is no extra charge for enabling server access logging on an Amazon S3 bucket, and you are not charged when the logs are PUT to your bucket. However, any log files that the system delivers to your bucket accrue the usual charges for storage. You can delete these log files at any time. Subsequent reads and other requests to these log files are charged normally, as for any other object, including data transfer charges.

By default, logging is disabled. When logging is enabled, logs are saved to a bucket in the same AWS Region as the source bucket.

This is the right choice since it does not add any additional costs while helping trace the user requests made on Amazon S3 buckets.

Incorrect options:

Use AWS X-Ray for tracing Amazon S3 requests from end-to-end - AWS X-Ray collects data about requests that your application serves. You can then view and filter the data to identify and troubleshoot performance issues and errors in your distributed applications and micro-services architecture. For any traced request to your application, it shows you detailed information about the request, the response, and the calls that your application makes to downstream AWS resources, micro-services, databases, and HTTP web APIs.

AWS X-Ray is overkill for the current requirement and also incurs extra costs since X-Ray is a paid service.

Use Amazon Inspector, a security assessment service that helps track the calls made to AWS services configured with it - Amazon Inspector security assessments help you check for unintended network accessibility of your Amazon EC2 instances and vulnerabilities on those EC2 instances. This service is not for tracing Amazon S3 access requests.

Use AWS CloudTrail to identify the requests made to the Amazon S3 buckets - Amazon S3 lets you identify requests using an AWS CloudTrail event log. By default, CloudTrail logs S3 bucket-level API calls that were made in the last 90 days, but not log requests made to objects.

AWS suggests the use of CloudTrail to access information on Amazon S3 buckets. However, the data needed for the current requirement is also provided by S3 server access logging, which is a free service.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/using-s3-access-logs-to-identify-requests.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/cloudtrail-request-identification.html#cloudtrail-identification-object-access

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

Question 36: **Skipped**

An organization has multiple AWS accounts to manage different lines of business. A user from the Finance account has to access reports stored in Amazon S3 buckets of two other AWS accounts (belonging to the HR and Audit departments) and copy these reports back to the S3 bucket in the Finance account. The user has requested the necessary permissions from the systems administrator to perform this task.

As a SysOps Administrator, how will you configure a solution for this requirement?

○ **Create resource-based policies in the HR, Audit accounts that will allow the requester from the Finance account to access the respective S3 buckets**

○ **Create IAM roles in the HR, Audit accounts, which can be assumed by the user from the Finance account when the user needs to access the S3 buckets of the accounts**

○ **Create resource-level permissions in the HR, Audit accounts to allow access to respective S3 buckets for the user in the Finance account**

○ **Create identity-based IAM policy in the Finance account that allows the user to make a request to the S3 buckets in the HR and Audit accounts. Also, create resource-based IAM policies in the HR, Audit accounts that will allow the requester from the Finance account to access the respective S3 buckets** **(Correct)**

Explanation

Correct option:

Create identity-based IAM policy in the Finance account that allows the user to make a request to the S3 buckets in the HR and Audit accounts. Also, create resource-based IAM policies in the HR, Audit accounts that will allow the requester from the Finance account to access the respective S3 buckets

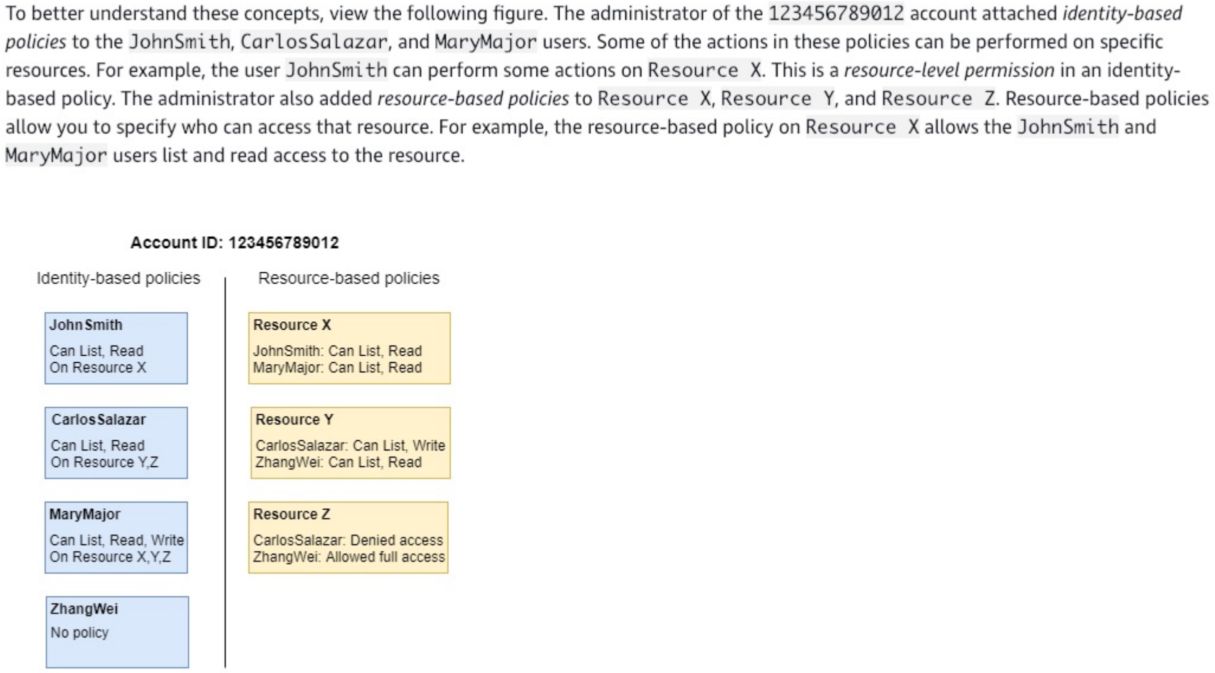
Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions).

Resource-based policies are attached to a resource. For example, you can attach resource-based policies to Amazon S3 buckets, Amazon SQS queues, and AWS Key Management Service encryption keys.

Identity-based policies and resource-based policies are both permissions policies and are evaluated together. For a request to which only permissions policies apply, AWS first checks all policies for a Deny. If one exists, then the request is denied. Then AWS checks for each Allow. If at least one policy statement allows the action in the request, the request is allowed. It doesn't matter whether the Allow is in the identity-based policy or the resource-based policy.

For requests made from one account to another, the requester in Account A must have an identity-based policy that allows them to make a request to the resource in Account B. Also, the resource-based policy in Account B must allow the requester in Account A to access the resource. There must be policies in both accounts that allow the operation, otherwise, the request fails.

Comparing IAM policies:



To better understand these concepts, view the following figure. The administrator of the 123456789012 account attached *identity-based policies* to the JohnSmith, CarlosSalazar, and MaryMajor users. Some of the actions in these policies can be performed on specific resources. For example, the user JohnSmith can perform some actions on Resource X. This is a *resource-level permission* in an identity-based policy. The administrator also added *resource-based policies* to Resource X, Resource Y, and Resource Z. Resource-based policies allow you to specify who can access that resource. For example, the resource-based policy on Resource X allows the JohnSmith and MaryMajor users list and read access to the resource.

Account ID: 123456789012

Identity-based policies	Resource-based policies
JohnSmith Can List, Read On Resource X	Resource X JohnSmith: Can List, Read MaryMajor: Can List, Read
CarlosSalazar Can List, Read On Resource Y,Z	Resource Y CarlosSalazar: Can List, Write ZhangWei: Can List, Read
MaryMajor Can List, Read, Write On Resource X,Y,Z	Resource Z CarlosSalazar: Denied access ZhangWei: Allowed full access
ZhangWei No policy	

via - https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html

Incorrect options:

Create resource-based policies in the HR, Audit accounts that will allow the requester from the Finance account to access the respective S3 buckets - Creating resource-based policy alone will be sufficient when the request is made within a single AWS account.

Create resource-level permissions in the HR, Audit accounts to allow access to respective S3 buckets for the user in the Finance account - Resource-based policies differ from resource-level permissions. You can attach resource-based policies directly to a resource, as described in this topic. Resource-level permissions refer to the ability to use ARNs to specify individual resources in a policy. Resource-based policies are supported only by some AWS services.

Create IAM roles in the HR, Audit accounts, which can be assumed by the user from the Finance account when the user needs to access the S3 buckets of the accounts - Cross-account access with a resource-based policy has some advantages over cross-account access with a role. With a resource that is accessed through a resource-based policy, the principal still works in the trusted account and does not have to give up his or her permissions to receive the role permissions. In other words, the principal continues to have access to resources in the trusted account at the same time as he or she has access to the resource in the trusting account. This is useful for tasks such as copying information to or from the shared resource in the other account.

We chose resource-based policy, so the user from the Finance account will continue to have access to resources in his own account while also getting permissions on resources from other accounts.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_identity-vs-resource.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_compare-resource-policies.html

Question 37: **Incorrect**

An e-commerce company runs its web application on Amazon EC2 instances backed by Amazon Elastic Block Store (Amazon EBS) volumes. An Amazon S3 bucket is used for storing sharable data. A developer has attached an Amazon EBS to an Amazon EC2 instance, but it’s still in the "attaching" state after 10-15 minutes.

As a SysOps Administrator, what solution will you suggest to fix this issue with the EBS volume?

○ **Check that the device name you specified when you attempted to attach the EBS volume isn't already in use. Attempt to attach the volume to the instance, again, but use a different device name** **(Correct)**

○ **The EBS volume could be encrypted and the custom KMS key used to encrypt the snapshot is missing. The custom KMS key needs to be added to the volume configuration**

○ **The `attaching` status indicates that the underlying hardware related to your EBS volume has failed. This issue cannot be fixed. Raise a service request on AWS and request for a new volume. You are not charged for volumes that are in error state** **(Incorrect)**

○ **Each EBS volume receives an initial I/O credit balance, an error in accumulating the credit balance can stop the volume from attaching properly to the instance. Restart the instance to fix the error**

Explanation

Correct option:

Check that the device name you specified when you attempted to attach the EBS volume isn't already in use. Attempt to attach the volume to the instance, again, but use a different device name

Check that the device name you specified when you attempted to attach the EBS volume isn't already in use. If the specified device name is already being used by the block device driver of the EC2 instance, the operation fails.

When attaching an EBS volume to an Amazon EC2 instance, you can specify a device name for the volume (by default, one is filled in for you). The block device driver of the EC2 instance mounts the volume and assigns a name. The volume name can be different from the name that you assign.

If you specify a device name that's not in use by Amazon EC2, but is used by the block device driver within the EC2 instance, the attachment of the Amazon EBS volume fails. Instead, the EBS volume is stuck in the attaching state. This is usually due to one of the following reasons:

1. The block device driver is remapping the specified device name: On an HVM EC2 instance, /dev/sda1 remaps to /dev/xvda. If you attempt to attach a secondary Amazon EBS volume to /dev/xvda, the secondary EBS volume can't successfully attach to the instance. This can cause the EBS volume to be stuck in the attaching state.
2. The block device driver didn't release the device name: If a user has initiated a forced detach of an Amazon EBS volume, the block device driver of the Amazon EC2 instance might not immediately release the device name for reuse. Attempting to use that device name when attaching a volume causes the volume to be stuck in the attaching state. You must either choose a different device name or reboot the instance.

You can resolve most issues with volumes stuck in the attaching state by following these steps: Force detach the volume and attempt to attach the volume to the instance, again, but use a different device name. The instance must be in running state for this to work.

If the above does not solve the problem, you can reboot the instance or stop and start the instance to migrate it to new underlying hardware. Keep in mind that instance store data is lost when you stop and start an instance.

Incorrect options:

The EBS volume could be encrypted and the custom KMS key used to encrypt the snapshot is missing. The custom KMS key needs to be added to the volume configuration - A missing KMS key will not lead to `attaching` state of the volume.

Each EBS volume receives an initial I/O credit balance, an error in accumulating the credit balance can stop the volume from attaching properly to the instance. Restart the instance to fix the error - This is a made-up option and has been added as a distractor.

The `attaching` status indicates that the underlying hardware related to your EBS volume has failed. This issue cannot be fixed. Raise a service request on AWS and request for a new volume. You are not charged for volumes that are in error state - When the underlying hardware related to your EBS volume has failed, the EBS volume will have a status of `error`. The data associated with the volume is unrecoverable and Amazon EBS processes the volume as lost. AWS doesn't bill for volumes with a status of `error`.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/ebs-stuck-attaching/

https://docs.amazonaws.cn/en_us/AWSEC2/latest/WindowsGuide/ebs-volume-types.html

Question 38: **Correct**

An e-commerce company uses AWS Elastic Beanstalk to create test environments comprising of an Amazon EC2 instance and an RDS instance whenever a new product or line-of-service is launched. The company is currently testing one such environment but wants to decouple the database from the environment to run some analysis and reports later in another environment. Since testing is in progress for a high-stakes product, the company wants to avoid downtime and database sync issues.

As a SysOps Administrator, which solution will you recommend to the company?

○ **Use an Elastic Beanstalk Immutable deployment to make the entire architecture completely reliable. You can terminate the first environment whenever you are confident of the second environment working correctly**

○ **Use an Elastic Beanstalk blue (environment A)/green (environment B) deployment to decouple the RDS DB instance from environment A. Create a new Elastic Beanstalk environment (environment B) with the necessary information to connect to the decoupled RDS DB instance** **(Correct)**

○ **Decoupling an RDS instance that is part of a running Elastic Beanstalk environment is not currently supported by AWS. You will need to terminate the current environment after taking the snapshot of the database and create a new one with RDS configured outside the environment**

○ **Since it is a test environment, take a snapshot of the database and terminate the current environment. Create a new one without attaching an RDS instance directly to it (from the snapshot)**

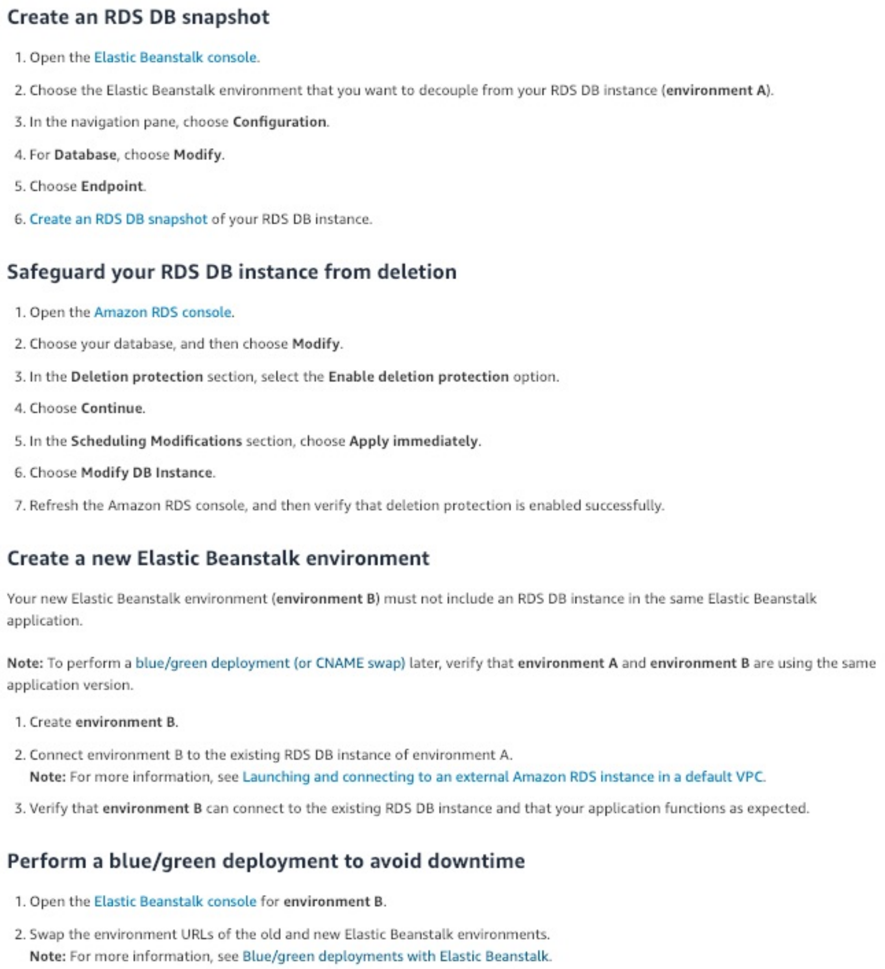
Explanation

Correct option:

Use an Elastic Beanstalk blue (environment A)/green (environment B) deployment to decouple the RDS DB instance from environment A. Create a new Elastic Beanstalk environment (environment B) with the necessary information to connect to the decouple RDS DB instance - Attaching an RDS DB instance to an Elastic Beanstalk environment is ideal for development and testing environments. However, it's not recommended for production environments because the lifecycle of the database instance is tied to the lifecycle of your application environment. If you terminate the environment, then you lose your data because the RDS DB instance is deleted by the environment.

Since the current use case mentions not having downtime on the database, we can follow these steps for resolution: 1. Use an Elastic Beanstalk blue (environment A)/green (environment B) deployment to decouple an RDS DB instance from environment A. Create an RDS DB snapshot and enable `Deletion protection` on the DB instance to Safeguard your RDS DB instance from deletion. 2. Create a new Elastic Beanstalk environment (environment B) with the necessary information to connect to the RDS DB instance. Your new Elastic Beanstalk environment (environment B) must not include an RDS DB instance in the same Elastic Beanstalk application.

Step-by-step instructions to configure the above solution:



Create an RDS DB snapshot

1. Open the Elastic Beanstalk console.
2. Choose the Elastic Beanstalk environment that you want to decouple from your RDS DB instance (**environment A**).
3. In the navigation pane, choose **Configuration**.
4. For **Database**, choose **Modify**.
5. Choose **Endpoint**.
6. Create an RDS DB snapshot of your RDS DB instance.

Safeguard your RDS DB instance from deletion

1. Open the Amazon RDS console.
2. Choose your database, and then choose **Modify**.
3. In the **Deletion protection** section, select the **Enable deletion protection** option.
4. Choose **Continue**.
5. In the **Scheduling Modifications** section, choose **Apply immediately**.
6. Choose **Modify DB Instance**.
7. Refresh the Amazon RDS console, and then verify that deletion protection is enabled successfully.

Create a new Elastic Beanstalk environment

Your new Elastic Beanstalk environment (**environment B**) must not include an RDS DB instance in the same Elastic Beanstalk application.

Note: To perform a blue/green deployment (or CNAME swap) later, verify that **environment A** and **environment B** are using the same application version.

1. Create **environment B**.
2. Connect environment B to the existing RDS DB instance of environment A.
   Note: For more information, see Launching and connecting to an external Amazon RDS instance in a default VPC.
3. Verify that **environment B** can connect to the existing RDS DB instance and that your application functions as expected.

Perform a blue/green deployment to avoid downtime

1. Open the Elastic Beanstalk console for **environment B**.
2. Swap the environment URLs of the old and new Elastic Beanstalk environments.
   Note: For more information, see Blue/green deployments with Elastic Beanstalk.

via

- https://aws.amazon.com/premiumsupport/knowledge-center/decouple-rds-from-beanstalk/

Incorrect options:

Since it is a test environment, take a snapshot of the database and terminate the current environment. Create a new one without attaching an RDS instance directly to it (from the snapshot) - It is mentioned in the problem statement that the company is looking at a solution with no downtime. Hence, this is an incorrect option.

Use an Elastic Beanstalk Immutable deployment to make the entire architecture completely reliable. You can terminate the first environment whenever you are confident of the second environment working correctly - Immutable deployments perform an immutable update to launch a full set of new instances running the new version of the application in a separate Auto Scaling group, alongside the instances running the old version. Immutable deployments can prevent issues caused by partially completed rolling deployments. If the new instances don't pass health checks, Elastic Beanstalk terminates them, leaving the original instances untouched. This solution is an over-kill for the test environment, even if the company is looking at a no-downtime option.

Decoupling an RDS instance that is part of a running Elastic Beanstalk environment is not currently supported by AWS. You will need to terminate the current environment after taking the snapshot of the database and create a new one with RDS configured outside the environment - This is a made-up option and given only as a distractor.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/decouple-rds-from-beanstalk/

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

Question 39: **Incorrect**

A junior developer created multiple stacks of resources in different AWS Regions per the CloudFormation template given to him. The development team soon started having issues with the created resources and their behavior. Initial checks have confirmed that some resources were created and some omitted, though the same template has been used. As a SysOps Administrator, you have been tasked to resolve these issues.

Which of the following could be the possible reason for this unexpected behavior?

○ **There might have been dependency errors, that resulted in the stack not being created completely**

○ **The CloudFormation template might have custom named IAM resources that are responsible for the unintended behavior** (Correct)

○ **The CloudFormation template was created using `use-once only` option and is not supposed to be reused for creating other stacks**

○ **Insufficient IAM permissions can lead to issues. When you work with an AWS CloudFormation stack, you not only need permissions to use AWS CloudFormation, you must also have permission to use the underlying services that are described in your template** (Incorrect)

Explanation

Correct option:

The CloudFormation template might have custom named IAM resources that are responsible for the unintended behavior - If your template contains custom named IAM resources, don't create multiple stacks reusing the same template. IAM resources must be globally unique within your account. If you use the same template to create multiple stacks in different Regions, your stacks might share the same IAM resources, instead of each having a unique one. Shared resources among stacks can have unintended consequences from which you can't recover. For example, if you delete or update shared IAM resources in one stack, you will unintentionally modify the resources of other stacks.

Incorrect options:

There might have been dependency errors, that resulted in the stack not being created completely - Any error during stack creation will rollback the entire stack creation process and the result is, none of the mentioned resources are created.

Insufficient IAM permissions can lead to issues. When you work with an AWS CloudFormation stack, you not only need permissions to use AWS CloudFormation, you must also have permission to use the underlying services that are described in your template - If permissions were an issue, the stack wouldn't be created at all.

The CloudFormation template was created using `use-once only` option and is not supposed to be reused for creating other stacks - This is a made-up option and given only as a distractor.

Reference:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-iam-template.html

Question 40: **Incorrect**

A junior systems administrator has created read replicas for Amazon RDS for MYSQL. The created read replicas are running into errors consistently.

As a SysOps Administrator, which of the following items would you suggest while troubleshooting read replica errors? (Select two)

Statements containing non-deterministic functions like SYSDATE() should be predefined in the configuration to successfully create the read replica

Writing to tables on a read replica can break the replication (Correct)

If the value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance, replica errors occur (Correct)

Though read replicas can work on both transactional and nontransactional storage engines, nontransactional engines are error-prone because of the way memory is managed on these engines (Incorrect)

To safely write to tables on a read replica, create indexes on the table after setting the read_only parameter to 0

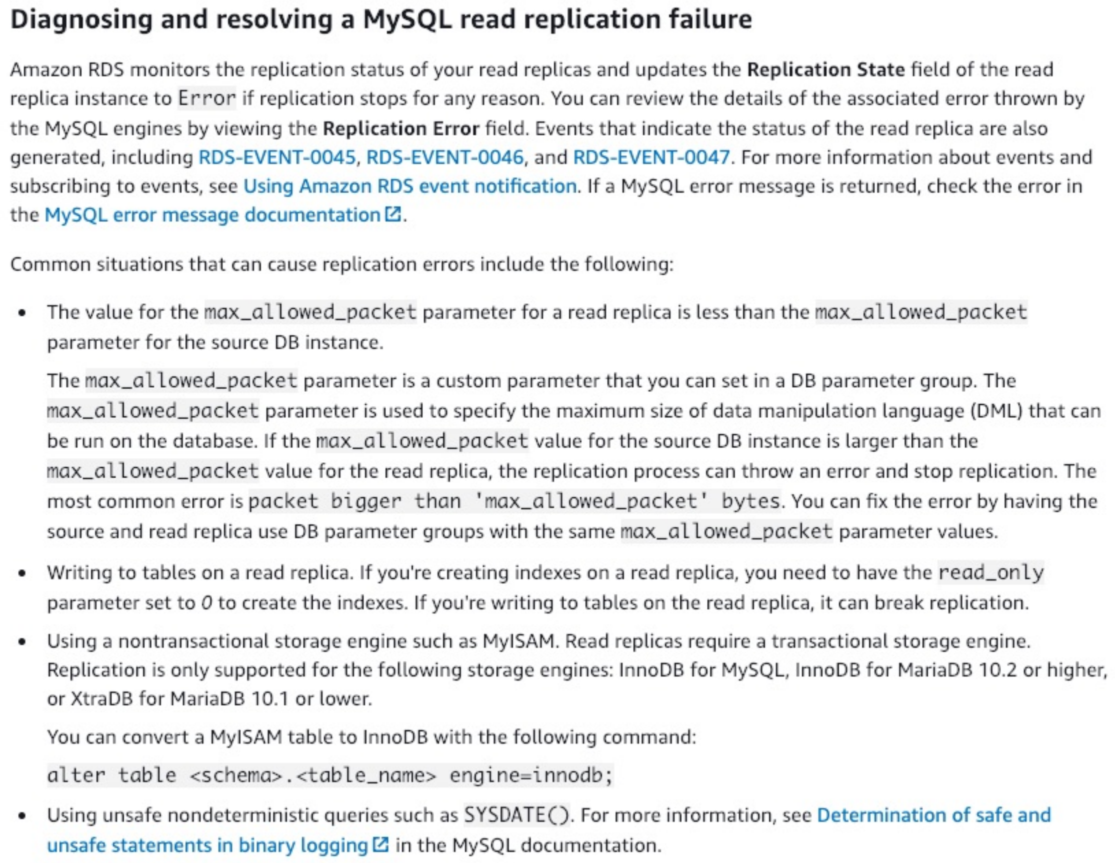
Explanation

Correct option:

Writing to tables on a read replica can break the replication - If you're writing to tables on the read replica, it can break replication.

If the value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance, replica errors occur - The max_allowed_packet parameter is a custom parameter that you can set in a DB parameter group. The max_allowed_packet parameter is used to specify the maximum size of data manipulation language (DML) that can be run on the database. If the max_allowed_packet value for the source DB instance is larger than the max_allowed_packet value for the read replica, the replication process can throw an error and stop replication.

Diagnosing MySQL read replication failure:



Diagnosing and resolving a MySQL read replication failure

Amazon RDS monitors the replication status of your read replicas and updates the **Replication State** field of the read replica instance to `Error` if replication stops for any reason. You can review the details of the associated error thrown by the MySQL engines by viewing the **Replication Error** field. Events that indicate the status of the read replica are also generated, including RDS-EVENT-0045, RDS-EVENT-0046, and RDS-EVENT-0047. For more information about events and subscribing to events, see Using Amazon RDS event notification. If a MySQL error message is returned, check the error in the MySQL error message documentation.

Common situations that can cause replication errors include the following:

- The value for the `max_allowed_packet` parameter for a read replica is less than the `max_allowed_packet` parameter for the source DB instance.

  The `max_allowed_packet` parameter is a custom parameter that you can set in a DB parameter group. The `max_allowed_packet` parameter is used to specify the maximum size of data manipulation language (DML) that can be run on the database. If the `max_allowed_packet` value for the source DB instance is larger than the `max_allowed_packet` value for the read replica, the replication process can throw an error and stop replication. The most common error is `packet bigger than 'max_allowed_packet' bytes`. You can fix the error by having the source and read replica use DB parameter groups with the same `max_allowed_packet` parameter values.
- Writing to tables on a read replica. If you're creating indexes on a read replica, you need to have the `read_only` parameter set to *0* to create the indexes. If you're writing to tables on the read replica, it can break replication.
- Using a nontransactional storage engine such as MyISAM. Read replicas require a transactional storage engine. Replication is only supported for the following storage engines: InnoDB for MySQL, InnoDB for MariaDB 10.2 or higher, or XtraDB for MariaDB 10.1 or lower.

  You can convert a MyISAM table to InnoDB with the following command:

  ```
  alter table <schema>.<table_name> engine=innodb;
  ```
- Using unsafe nondeterministic queries such as `SYSDATE()`. For more information, see Determination of safe and unsafe statements in binary logging in the MySQL documentation.

via

- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.MySQL.ReplicaLag

Incorrect options:

To safely write to tables on a read replica, create indexes on the table after setting the read_only parameter to 0 - As discussed above, writing to tables in read replica breaks the replication process. Setting the read_only paramater to 0 will not help.

Though read replicas can work on both transactional and nontransactional storage engines, nontransactional engines are error-prone because of the way memory is managed on these engines - Read replicas can only work on a transactional storage engine. Using a nontransactional storage engine such as MyISAM can break the replication process.

Statements containing non-deterministic functions like SYSDATE() should be predefined in the configuration to successfully create the read replica - This statement is incorrect. Using unsafe nondeterministic queries such as SLEEP(), SYSDATE(), SYSTEM_USER(), etc can break the replication. There is no option to predefine such functions in the configuration.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_Troubleshooting.html#CHAP_Troubleshooting.MySQL.ReplicaLag

https://dev.mysql.com/doc/refman/8.0/en/replication-rbr-safe-unsafe.html

https://aws.amazon.com/premiumsupport/knowledge-center/rds-read-replica/

Question 41: **Correct**

The technology team at a retail company has set the `DisableApiTermination` attribute for a business-critical Amazon EC2 Windows instance to prevent termination of the instance via an API. This instance is behind an Auto Scaling Group (ASG) and the `InstanceInitiatedShutdownBehavior` attribute is set for the instance. A developer has initiated shutdown from the instance using operating system commands.

What will be the outcome of the above scenario?

○ **The operating system of the instance will send an Amazon SNS notification to the concerned person, that was configured when `DisableApiTermination` attribute was set. The operating system will hold the shutdown for few configured minutes and then progress with instance shutdown**

○ **ASG cannot terminate an instance whose `DisableApiTermination` attribute is set**

○ **The instance will not shutdown because `DisableApiTermination` attribute is set**

○ **The instance will be terminated** **(Correct)**

Explanation

Correct option:

The instance will be terminated - 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. The DisableApiTermination attribute controls whether the instance can be terminated using the console, CLI, or API. By default, termination protection is disabled for your instance. You can set the value of this attribute when you launch the instance, while the instance is running, or while the instance is stopped (for Amazon EBS-backed instances).

The DisableApiTermination attribute does not prevent you from terminating an instance by initiating shutdown from the instance (using an operating system command for system shutdown) when the `InstanceInitiatedShutdownBehavior` attribute is set.

Incorrect options:

The instance will not shutdown because `DisableApiTermination` attribute is set - As discussed above, this flag is only for controlling instance termination from console, command line interface, or API. If does not protect from shutdown commands issued from the operating system of the instance if the `InstanceInitiatedShutdownBehavior` attribute is set.

The operating system of the instance will send an Amazon SNS notification to the concerned person, that was configured when `DisableApiTermination` attribute was set. The operating system will hold the shutdown for few configured minutes and then progress with instance shutdown - This is a made-up option and given only as a distractor.

ASG cannot terminate an instance whose `DisableApiTermination` attribute is set - This statement is false. The DisableApiTermination attribute does not prevent Amazon EC2 Auto Scaling from terminating an instance.

Reference:

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

Question 42: **Correct**

A large online business uses multiple Amazon EBS volumes for their storage requirements. According to the company guidelines, the EBS snapshots have to be taken every few minutes to retain the business-critical data in case of failure.

As a SysOps Administrator, can you suggest an effective way of addressing this requirement?

○ **Automated EBS snapshots is a configurable item from Amazon EC2 configuration screen on AWS console**

○ **Use Amazon CloudWatch events to schedule automated EBS Snapshots** **(Correct)**

○ **Use Amazon SNS Notification service to trigger AWS Lambda function that can initiate the EBS snapshots**

○ **Use AWS Lambda functions to initiate automatic EBS snapshots every few minutes**

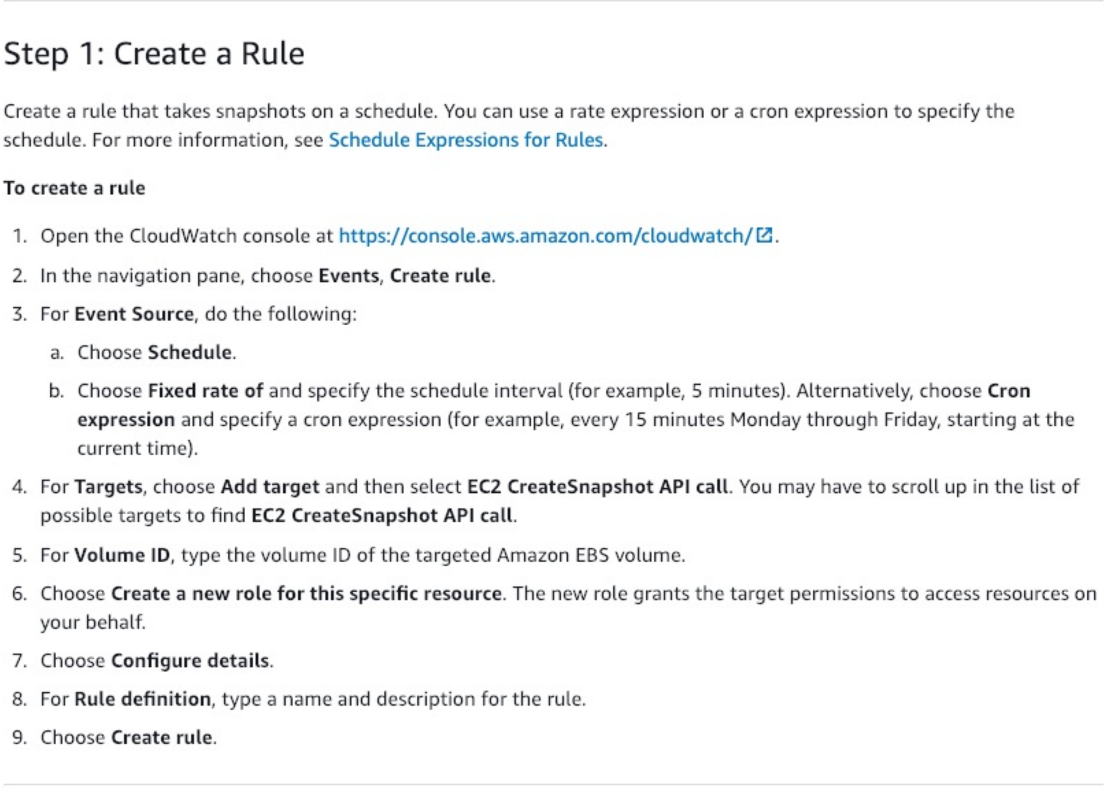
Explanation

Correct option:

Use Amazon CloudWatch events to schedule automated EBS Snapshots - You can run CloudWatch Events rules according to a schedule. It is possible to create an automated snapshot of an existing Amazon Elastic Block Store (Amazon EBS) volume on a schedule. You can choose a fixed rate to create a snapshot every few minutes or use a cron expression to specify that the snapshot is made at a specific time of day.

Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

Steps to create a rule that takes snapshots on a schedule:



Step 1: Create a Rule

Create a rule that takes snapshots on a schedule. You can use a rate expression or a cron expression to specify the schedule. For more information, see Schedule Expressions for Rules.

To create a rule

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Events**, **Create rule**.
3. For **Event Source**, do the following:
   a. Choose **Schedule**.
   b. Choose **Fixed rate of** and specify the schedule interval (for example, 5 minutes). Alternatively, choose **Cron expression** and specify a cron expression (for example, every 15 minutes Monday through Friday, starting at the current time).
4. For **Targets**, choose **Add target** and then select **EC2 CreateSnapshot API call**. You may have to scroll up in the list of possible targets to find **EC2 CreateSnapshot API call**.
5. For **Volume ID**, type the volume ID of the targeted Amazon EBS volume.
6. Choose **Create a new role for this specific resource**. The new role grants the target permissions to access resources on your behalf.
7. Choose **Configure details**.
8. For **Rule definition**, type a name and description for the rule.
9. Choose **Create rule**.

via - https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/TakeScheduledSnapshot.html

Incorrect options:

Use AWS Lambda functions to initiate automatic EBS snapshots every few minutes - AWS Lambda is not a self-invoking function and needs a service to invoke it. Writing code to self invoke in the same Lambda function will result in too many parallel invocations and will turn out to be a very expensive solution. Hence, this option is incorrect.

Use Amazon SNS Notification service to trigger AWS Lambda function that can initiate the EBS snapshots - Amazon SNS and AWS Lambda are integrated so you can invoke Lambda functions with Amazon SNS notifications. Lambda function can be coded to take a snapshot of EBS volume every few minutes. However, this process is neither direct nor is cost-effective way of achieving the stated requirement.

Automated EBS snapshots is a configurable item from Amazon EC2 configuration screen on AWS console - This is an incorrect statement and given only as a distractor.

Reference:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/TakeScheduledSnapshot.html

Question 43: **Incorrect**

As a SysOps Administrator, you have been contacted by a team for troubleshooting a security issue they seem to be facing. A security check red flag is being raised for the security groups created by AWS Directory Services. The flag message says "Security Groups -

Unrestricted Access."

How will you troubleshoot this issue?

○ **The security group configurations have to be checked and edited to cater to AWS security standards**

○ **AWS Directory Service might have been initiated from an account that does not have proper permissions. Check the permissions on the IAM roles and IAM users used to initiate the service** **(Incorrect)**

○ **Ignore or suppress the red flag since it is safe to do so, in this scenario** **(Correct)**

○ **Use AWS Trusted Advisor to know the exact reason for this error and take action as recommended by the Trusted Advisor**

Explanation

Correct option:

Ignore or suppress the red flag since it is safe to do so, in this scenario - AWS Directory Services is a managed service that automatically creates an AWS security group in your VPC with network rules for traffic in and out of AWS managed domain controllers. The default inbound rules allow traffic from any source (0.0.0.0/0) to ports required by Active Directory. These rules do not introduce security vulnerabilities, as traffic to the domain controllers is limited to traffic from your VPC, other peered VPCs, or networks connected using AWS Direct Connect, AWS Transit Gateway or Virtual Private Network.

In addition, the ENIs the security group is attached to, do not and cannot have Elastic IPs attached to them, limiting inbound traffic to local VPC and VPC routed traffic.

Incorrect options:

The security group configurations have to be checked and edited to cater to AWS security standards

Use AWS Trusted Advisor to know the exact reason for this error and take action as recommended by the Trusted Advisor

AWS Directory Service might have been initiated from an account that does not have proper permissions. Check the permissions on the IAM roles and IAM users used to initiate the service

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

Reference:

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

Question 44: **Correct**

An automobile company manages its AWS resource creation and maintenance process through AWS CloudFormation. The company has successfully used CloudFormation so far, and wishes to continue using the service. However, while moving to CloudFormation, the company only moved critical resources and left out the other resources to be managed manually. To leverage the ease of creation and maintenance that CloudFormation offers, the company wants to move rest of the resources to CloudFormation.

Which of the following options is the recommended way to configure this requirement?

○ **Drift detection is the mechanism by which you add resources to the stack of Cloudformation resources already created**

○ **You can use `Mappings` part of CloudFormation template to input the needed resources**

○ **Use `Parameters` section of CloudFormation template to input the required resources**

○ **You can bring an existing resource into AWS CloudFormation management using `resource import`** **(Correct)**

Explanation

Correct option:

You can bring an existing resource into AWS CloudFormation management using `resource import`

If you created an AWS resource outside of AWS CloudFormation management, you can bring this existing resource into AWS CloudFormation management using `resource import`. You can manage your resources using AWS CloudFormation regardless of where they were created without having to delete and re-create them as part of a stack.

During an import operation, you create a change set that imports your existing resources into a stack or creates a new stack from your existing resources. You provide the following during import.

1. A template that describes the entire stack, including both the original stack resources and the resources you're importing. Each resource to import must have a DeletionPolicy attribute.
2. Identifiers for the resources to import. You provide two values to identify each target resource.

a) An identifier property. This is a resource property that can be used to identify each resource type. For example, an AWS::S3::Bucket resource can be identified using its BucketName.

b) An identifier value. This is the target resource's actual property value. For example, the actual value for the BucketName property might be MyS3Bucket.

Incorrect options:

Use `Parameters` section of CloudFormation template to input the required resources - Parameters are a way to provide inputs to your AWS CloudFormation template. They are useful when you want to reuse your templates. Some inputs can not be determined ahead of time. They aren't useful for importing resources into CloudFormation.

You can use `Mappings` part of CloudFormation template to input the needed resources - Mappings are fixed variables within your CloudFormation Template. They're very handy to differentiate between different environments (dev vs prod), regions (AWS regions), AMI types, etc. They aren't useful for importing resources into CloudFormation.

Drift detection is the mechanism by which you add resources to the stack of Cloudformation resources already created - Performing a drift detection operation on a stack determines whether the stack has drifted from its expected template configuration, and returns detailed information about the drift status of each resource in the stack that supports drift detection. It is not useful for importing resources into CloudFormation.

Reference:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resource-import.html

Question 45: **Correct**

The Chief Technology Officer (CTO) of a healthcare company realized that he does not have access to an Amazon S3 bucket present in the company's own AWS account. The CTO is the root user for the AWS account and has created other AWS users using the root user account.

What is the reason for this behavior and how can you fix this?

○ **An Amazon S3 bucket policy that specifies a wildcard (*) in the principal element, sometimes is declared void by AWS to avoid the risk of complete public exposure. Such S3 buckets policies are in invalid status and have random behavior**

○ **Root user always has access to all the resources of the account. The Amazon S3 bucket could be from another AWS account and the S3 bucket has been shared with the root user and hence appears in his list of S3 buckets**

○ **Root user has access to all the resources in his AWS account. Contact AWS support to resolve the access issue**

○ **If an IAM user, with full access to IAM and Amazon S3, assigns a bucket policy to an Amazon S3 bucket and doesn't specify the AWS account root user as a principal, the root user is denied access to that bucket** **(Correct)**

Explanation

Correct option:

If an IAM user, with full access to IAM and Amazon S3, assigns a bucket policy to an Amazon S3 bucket and doesn't specify the AWS account root user as a principal, the root user is denied access to that bucket

Sometimes, you might have an IAM user with full access to IAM and Amazon S3. If the IAM user assigns a bucket policy to an Amazon S3 bucket and doesn't specify the AWS account root user as a principal, the root user is denied access to that bucket. However, as the root user, you can still access the bucket. To do that, modify the bucket policy to allow root user access from the Amazon S3 console or the AWS CLI. Use the following principal, replacing 123456789012 with the ID of the AWS account.

```
"Principal": { "AWS": "arn:aws:iam::123456789012:root" }
```

Incorrect options:

Root user always has access to all the resources of the account. The Amazon S3 bucket could be from another AWS account and the S3 bucket has been shared with the root user and hence appears in his list of S3 buckets

An Amazon S3 bucket policy that specifies a wildcard (*) in the principal element, sometimes is declared void by AWS to avoid the risk of complete public exposure. Such S3 buckets policies are in invalid status and have random behavior

Root user has access to all the resources in his AWS account. Contact AWS support to resolve the access issue

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

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/troubleshoot_iam-s3.html

Question 46: **Incorrect**

A team noticed that it has accidentally deleted the AMI of Amazon EC2 instances belonging to the test environment. The team had configured backups via EBS snapshots for these instances.

Which of the following options would you suggest to recover/rebuild the accidentally deleted AMI? (Select two)

AWS Support retains backups of AMIs. Write to the support team to get help for recovering the lost AMI

Create a new AMI from Amazon EC2 instances that were launched before the deletion of AMI (Correct)

Recover the AMI from Amazon EBS snapshots that were created as backups before the deletion of AMI (Incorrect)

Recover the AMI from the current Amazon EC2 instances that were launched before the deletion of AMI

Create a new AMI from Amazon EBS snapshots that were created as backups (Correct)

Explanation

Correct options:

Create a new AMI from Amazon EBS snapshots that were created as backups

Create a new AMI from Amazon EC2 instances that were launched before the deletion of AMI

It isn't possible to restore or recover a deleted or deregistered AMI. However, you can create a new, identical AMI using one of the following:

1. Amazon Elastic Block Store (Amazon EBS) snapshots that were created as backups: When you delete or deregister an Amazon EBS-backed AMI, any snapshots created for the volume of the instance during the AMI creation process are retained. If you accidentally delete the AMI, you can launch an identical AMI using one of the retained snapshots.
2. Amazon Elastic Compute Cloud (Amazon EC2) instances that were launched from the deleted AMI: If you deleted the AMI and the snapshots are also deleted, then you can recover the AMI from any existing EC2 instances launched using the deleted AMI. Unless you have selected the `No reboot` option on the instance, performing this step will reboot the instance.

Incorrect options:

AWS Support retains backups of AMIs. Write to the support team to get help for recovering the lost AMI - For security and privacy reasons, AWS Support doesn't have visibility or access to customer data. If you don't have backups of your deleted AMI, AWS Support can't recover it for you.

Recover the AMI from the current Amazon EC2 instances that were launched before the deletion of AMI

Recover the AMI from Amazon EBS snapshots that were created as backups before the deletion of AMI

As discussed above, it is not possible to restore or recover a deleted or deregistered AMI. The only option is to create a new, identical AMI as discussed above.

Reference:

https://aws.amazon.com/premiumsupport/knowledge-center/recover-ami-accidentally-deleted-ec2/

Question 47: **Correct**

A systems administration team is configuring Amazon EC2 metrics that are sent to Amazon CloudWatch for monitoring purposes. The team is looking for a metric that can help them identify the processing power required to run an application on the selected instance.

Which of the below metric should be used for this requirement?

○ **`CPUUtilization` metric should be used to identify the processing power required** **(Correct)**

○ **`CPUProcessPower` metric should be used to identify the processing power required**

○ **`ResourceCount` is the correct metric to identify the processing power required**

○ **`CPUCreditUsage` metric should be used to identify the processing power required**

Explanation

Correct option: `CPUUtilization` **metric should be used to identify the processing power required** - `CPUUtilization` specifies the percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application on a selected instance. This metric is expressed in Percent.

Depending on the instance type, tools in your operating system can show a lower percentage than CloudWatch when the instance is not allocated a full processor core.

Incorrect options:

`CPUCreditUsage` **metric should be used to identify the processing power required** - This metric identifies the number of CPU credits spent by the instance for CPU utilization. CPU credit metrics are available at a five-minute frequency only. If you specify a period greater than five minutes, use the Sum statistic instead of the Average statistic. The units of this metric are Credits (vCPU-minutes).

`ResourceCount` **is the correct metric to identify the processing power required** - `ResourceCount` metric defines the number of the specified resources running in your account. The resources are defined by the dimensions associated with the metric.

`CPUProcessPower` **metric should be used to identify the processing power required** - `CPUProcessPower` is a made-up option, given only as a distractor.

Reference:

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

Question 48: **Correct**

A company has recently moved its server infrastructure to Amazon EC2 instances. The company needs to use CloudWatch metrics to track the state of each of the instances.

Which of the following is the right way to configure the instances for CloudWatch monitoring to work?

○ **Configure CloudWatch from AWS Console for the instances that need to be monitored by CloudWatch. AWS automatically installs and configure the agent for the mentioned instances**

○ **Install CloudWatch Agent on all the instances and attach necessary Security Groups to the EC2 instances to be able to run the CloudWatch agent**

○ **Install CloudWatch Agent on all the instances and attach an IAM role to the EC2 instances to be able to run the CloudWatch agent** **(Correct)**

○ **Install CloudWatch Agent on all the instances and attach an IAM user to the EC2 instances to be able to run the CloudWatch agent**

Explanation

Correct option:

Install CloudWatch Agent on all the instances and attach an IAM role to the EC2 instances to be able to run the CloudWatch agent - Access to AWS resources requires permissions. You create an IAM role, an IAM user, or both to grant permissions that the CloudWatch agent needs to write metrics to CloudWatch. If you're going to use the agent on Amazon EC2 instances, you must create an IAM role.

You must attach the CloudWatchAgentServerRole IAM role to the EC2 instance to be able to run the CloudWatch agent on the instance. This role enables the CloudWatch agent to perform actions on the instance.

Incorrect options:

Configure CloudWatch from AWS Console for the instances that need to be monitored by CloudWatch. AWS automatically installs and configure the agent for the mentioned instances - This is an incorrect statement. CloudWatch Agent has to be installed by the customer manually using the CLI on the instances.

Install CloudWatch Agent on all the instances and attach an IAM user to the EC2 instances to be able to run the CloudWatch agent - If you're going to use the CloudWatch agent on Amazon EC2 instances, you should create an IAM role.

Install CloudWatch Agent on all the instances and attach necessary Security Groups to the EC2 instances to be able to run the CloudWatch agent - CloudWatch agent gets the necessary permissions for collecting the metrics from EC2 instances using either an IAM role or an IAM user. Security Groups cannot be used for configuring CloudWatch agents.

Reference:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-EC2-Instance-fleet.html

Question 49: **Correct**

A hospitality company runs their applications on its on-premises infrastructure but stores the critical customer data on AWS Cloud using AWS Storage Gateway. At a recent audit, the company has been asked if the customer data is secure while in-transit and at rest in the Cloud.

What is the correct answer to the auditor's question? And what should the company change to meet the security requirements?

○ **AWS Storage Gateway uses SSL/TLS (Secure Socket Layers/Transport Layer Security) to encrypt data that is transferred between your gateway appliance and AWS storage. File and Volume Gateway data stored on Amazon S3 is encrypted. Tape Gateway data cannot be encrypted at-rest**

○ **AWS Storage Gateway uses IPsec to encrypt data that is transferred between your gateway appliance and AWS storage. All three Gateway types store data in encrypted form at-rest**

○ **AWS Storage Gateway uses IPsec to encrypt data that is transferred between your gateway appliance and AWS storage. File and Volume Gateway data stored on Amazon S3 is encrypted. Tape Gateway data cannot be encrypted at-rest**

○ **AWS Storage Gateway uses SSL/TLS (Secure Socket Layers/Transport Layer Security) to encrypt data that is transferred between your gateway appliance and AWS storage. By default, Storage Gateway uses Amazon S3-Managed Encryption Keys to server-side encrypt all data it stores in Amazon S3** **(Correct)**

Explanation

Correct option:

AWS Storage Gateway uses SSL/TLS (Secure Socket Layers/Transport Layer Security) to encrypt data that is transferred between your gateway appliance and AWS storage. By default, Storage Gateway uses Amazon S3-Managed Encryption Keys to server-side encrypt all data it stores in Amazon S3

AWS Storage Gateway uses SSL/TLS (Secure Socket Layers/Transport Layer Security) to encrypt data that is transferred between your gateway appliance and AWS storage. By default, Storage Gateway uses Amazon S3-Managed Encryption Keys (SSE-S3) to server-side encrypt all data it stores in Amazon S3. You have an option to use the Storage Gateway API to configure your gateway to encrypt data stored in the cloud using server-side encryption with AWS Key Management Service (SSE-KMS) customer master keys (CMKs).

File, Volume and Tape Gateway data is stored in Amazon S3 buckets by AWS Storage Gateway. Tape Gateway supports backing data to Amazon S3 Glacier apart from the standard storage.

Encrypting a file share: For a file share, you can configure your gateway to encrypt your objects with AWS KMS–managed keys by using SSE-KMS.

Encrypting a volume: For cached and stored volumes, you can configure your gateway to encrypt volume data stored in the cloud with AWS KMS–managed keys by using the Storage Gateway API.

Encrypting a tape: For a virtual tape, you can configure your gateway to encrypt tape data stored in the cloud with AWS KMS–managed keys by using the Storage Gateway API.

Incorrect options:

AWS Storage Gateway uses IPsec to encrypt data that is transferred between your gateway appliance and AWS storage. File and Volume Gateway data stored on Amazon S3 is encrypted. Tape Gateway data cannot be encrypted at-rest

AWS Storage Gateway uses IPsec to encrypt data that is transferred between your gateway appliance and AWS storage. All three Gateway types store data in encrypted form at-rest

There is no such thing as using IPSec for encrypting in-transit data between your gateway appliance and AWS storage. You need to use SSL/TLS for this. So both these options are incorrect.

AWS Storage Gateway uses SSL/TLS (Secure Socket Layers/Transport Layer Security) to encrypt data that is transferred between your gateway appliance and AWS storage. File and Volume Gateway data stored on Amazon S3 is encrypted. Tape Gateway data cannot be encrypted at-rest - For a virtual tape, you can configure your gateway to encrypt tape data stored in the cloud with AWS KMS–managed keys by using the Storage Gateway API. So this option is incorrect.

Reference:

https://docs.aws.amazon.com/storagegateway/latest/userguide/encryption.html

Question 50: **Incorrect**

An IT company runs its server infrastructure on Amazon EC2 instances configured in an Auto Scaling Group (ASG) fronted by an Elastic Load Balancer (ELB). For ease of deployment and flexibility in scaling, this AWS architecture is maintained via an Elastic Beanstalk environment. The Technology Lead of a project has requested to automate the replacement of unhealthy Amazon EC2 instances in the Elastic Beanstalk environment.

How will you configure a solution for this requirement?

○ **To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from EC2 to ELB by using a configuration file of your Beanstalk environment** **(Correct)**

○ **Modify the Auto Scaling Group from Amazon EC2 console directly to change the health check type to ELB** **(Incorrect)**

○ **Modify the Auto Scaling Group from Amazon EC2 console directly to change the health check type to EC2**

○ **To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from ELB to EC2 by using a configuration file of your Beanstalk environment**

Explanation

Correct option:

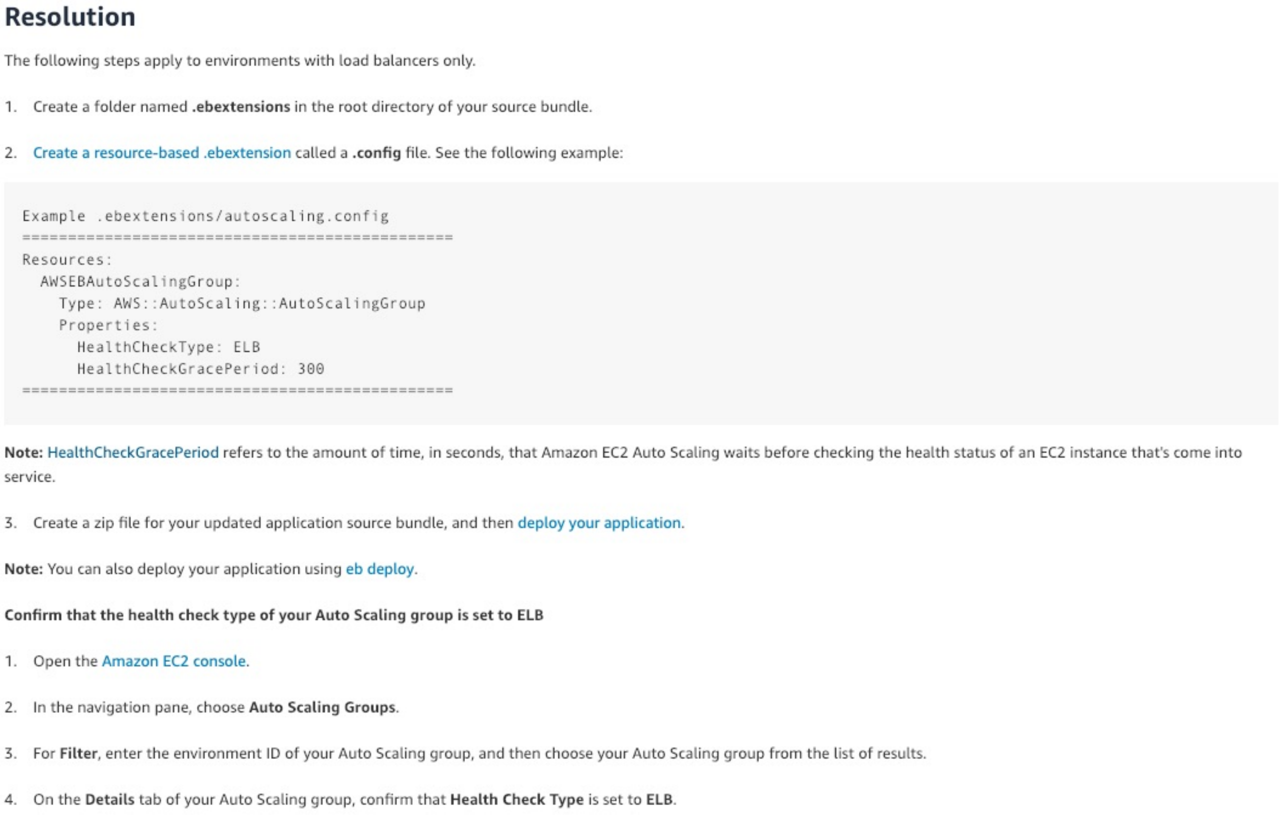
To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from EC2 to ELB by using a configuration file of your Beanstalk environment

By default, the health check configuration of your Auto Scaling group is set as an EC2 type that performs a status check of EC2 instances. To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from EC2 to ELB by using a configuration file.

The following are some important points to remember:

1. Status checks cover only an EC2 instance's health, and not the health of your application, server, or any Docker containers running on the instance.
2. If your application crashes, the load balancer removes the unhealthy instances from its target. However, your Auto Scaling group doesn't automatically replace the unhealthy instances marked by the load balancer.
3. By changing the health check type of your Auto Scaling group from EC2 to ELB, you enable the Auto Scaling group to automatically replace the unhealthy instances when the health check fails.

Complete list of steps to configure the above:



Resolution

The following steps apply to environments with load balancers only.

1. Create a folder named **.ebextensions** in the root directory of your source bundle.

2. Create a resource-based .ebextension called a **.config** file. See the following example:

```
Example .ebextensions/autoscaling.config
===============================================
Resources:
  AWSEBAutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
===============================================
```

Note: HealthCheckGracePeriod refers to the amount of time, in seconds, that Amazon EC2 Auto Scaling waits before checking the health status of an EC2 instance that's come into service.

3. Create a zip file for your updated application source bundle, and then deploy your application.

Note: You can also deploy your application using eb deploy.

Confirm that the health check type of your Auto Scaling group is set to ELB

1. Open the Amazon EC2 console.

2. In the navigation pane, choose **Auto Scaling Groups**.

3. For **Filter**, enter the environment ID of your Auto Scaling group, and then choose your Auto Scaling group from the list of results.

4. On the **Details** tab of your Auto Scaling group, confirm that **Health Check Type** is set to **ELB**.

via - https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-instance-automation/

Incorrect options:

To automate the replacement of unhealthy EC2 instances, you must change the health check type of your instance's Auto Scaling group from ELB to EC2 by using a configuration file of your Beanstalk environment - As mentioned earlier, the health check type of your instance's Auto Scaling group should be changed from EC2 to ELB.

Modify the Auto Scaling Group from Amazon EC2 console directly to change the health check type to ELB

Modify the Auto Scaling Group from Amazon EC2 console directly to change the health check type to EC2

You should configure your Amazon EC2 instances in an Elastic Beanstalk environment by using Elastic Beanstalk configuration files (.ebextensions). Configuration changes made to your Elastic Beanstalk environment won't persist if you use the following configuration methods:

1. Configuring an Elastic Beanstalk resource directly from the console of a specific AWS service.
2. Installing a package, creating a file, or running a command directly from your Amazon EC2 instance.

Both these options contradict the above explanation and therefore these two options are incorrect.

Reference:

https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-configuration-files/

Question 51: **Correct**

An e-commerce web application is built on a fleet of Amazon EC2 instances with an Auto Scaling Group. The application performance remains consistent throughout the day. But, for a few weeks now, users have been complaining about lagging screens and failing orders between 5-6 PM almost every day. Server logs show a sharp spike in user activity for this one hour every day.

What is an optimal way to fix the issue while keeping the application available?

○ **Create a scheduled scaling action to scale up before the traffic spike hits the servers** **(Correct)**

○ **You can choose to manually add few more instances to the ASG to deal with the sudden spike**

○ **Modify the Auto Scaling Group launch configuration to include more number of instances**

○ **Configure an Elastic Load Balancer, to replace the ASG, and move all the instances to ELB**

Explanation

Correct option:

Create a scheduled scaling action to scale up before the traffic spike hits the servers - Scheduled scaling allows you to set your own scaling schedule. For example, let's say that every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling actions based on the predictable traffic patterns of your web application. Scaling actions are performed automatically as a function of time and date.

To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. To create a scheduled scaling action, you specify the start time when the scaling action should take effect, and the new minimum, maximum, and desired sizes for the scaling action. At the specified time, Amazon EC2 Auto Scaling updates the group with the values for minimum, maximum, and desired size that are specified by the scaling action.

You can create scheduled actions for scaling one time only, or for scaling on a recurring schedule.

Incorrect options:

Modify the Auto Scaling Group launch configuration to include more number of instances - An Auto Scaling group is associated with one launch configuration at a time, and you can't modify a launch configuration after you've created it.

You can choose to manually add few more instances to the ASG to deal with the sudden spike - At any time, you can change the size of an existing Auto Scaling group manually. You can either update the desired capacity of the Auto Scaling group, or update the instances that are attached to the Auto Scaling group. But, this is not an optimal solution, since it requires user intervention on daily basis and an elegant and effective method is already available.

Configure an Elastic Load Balancer, to replace the ASG, and move all the instances to ELB - Elastic Load Balancer can balance the incoming traffic across instances. It cannot scale-out and launch new instances in the absence of an attached Auto Scaling Group.

References:

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

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-manual-scaling.html

Question 52: **Correct**

As SysOps Administrator, you have created two configuration files for CloudWatch Agent configuration. The first configuration file collects a set of metrics and logs from all servers and the second configuration file collects metrics from certain applications. You have given the same name to both the files but stored these files in different file paths.

What is the outcome when the CloudWatch Agent is started with the first configuration file and then the second configuration file is appended to it?

○ **Two different Agents are started with different configurations, collecting the metrics and logs listed in either of the configuration files**

○ **A CloudWatch Agent can have only one configuration file and all required parameters are defined in this file alone**

○ **The append command overwrites the information from the first configuration file instead of appending to it** **(Correct)**

○ **Second configuration file parameters are added to the Agent already running with the first configuration file parameters**

Explanation

Correct option:

The append command overwrites the information from the first configuration file instead of appending to it

You can set up the CloudWatch agent to use multiple configuration files. For example, you can use a common configuration file that collects a set of metrics and logs that you always want to collect from all servers in your infrastructure. You can then use additional configuration files that collect metrics from certain applications or in certain situations.

To set this up, first create the configuration files that you want to use. Any configuration files that will be used together on the same server must have different file names. You can store the configuration files on servers or in Parameter Store.

Start the CloudWatch agent using the `fetch-config` option and specify the first configuration file. To append the second configuration file to the running agent, use the same command but with the `append-config` option. All metrics and logs listed in either configuration file are collected.

Any configuration files appended to the configuration must have different file names from each other and from the initial configuration file. If you use `append-config` with a configuration file with the same file name as a configuration file that the agent is already using, the append command overwrites the information from the first configuration file instead of appending to it. This is true even if the two configuration files with the same file name are on different file paths.

Incorrect options:

Second configuration file parameters are added to the Agent already running with the first configuration file parameters

Two different Agents are started with different configurations, collecting the metrics and logs listed in either of the configuration files

A CloudWatch Agent can have only one configuration file and all required parameters are defined in this file alone

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

Reference:

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

Question 53: **Incorrect**

A firm uses Amazon EC2 instances for running its flagship application. With new business expansion plans, the firm is looking at a bigger footprint for its AWS infrastructure. The development team needs to share Amazon Machine Images (AMIs) across AZs, AWS accounts and Regions.

What are the key points to be considered before planning the expansion? (Select two)

You need to share any CMKs used to encrypt snapshots and any Amazon EBS snapshots that the AMI references (Incorrect)

You can only share AMIs that have unencrypted volumes and volumes that are encrypted with an AWS-managed CMK

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI (Correct)

You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer-managed CMK (Correct)

AMIs are regional resources and can be shared across Regions

Explanation

Correct options:

You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer-managed CMK - 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.

You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI - You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI. Only the AMI itself needs to be shared; the system automatically provides the access to the referenced Amazon EBS snapshots for the launch.

Incorrect options:

You can only share AMIs that have unencrypted volumes and volumes that are encrypted with an AWS-managed CMK - You cannot share an AMI that has volumes that are encrypted with an AWS-managed CMK.

You need to share any CMKs used to encrypt snapshots and any Amazon EBS snapshots that the AMI references - You do not need to share the Amazon EBS snapshots that an AMI references in order to share the AMI.

AMIs are regional resources and can be shared across Regions - 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. Sharing an AMI from different Regions is not available.

Reference:

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

Question 54: **Correct**

As a SysOps Administrator, you have been asked to fix the network performance issues for a fleet of Amazon EC2 instances of a company.

Which of the following use-cases represents the right fit for using enhanced networking?

○ **To configure Direct Connect to reach speeds up to 25 Gbps between EC2 instances**

○ **To configure multi-attach for an EBS volume that can be attached to a maximum of 16 EC2 instances in a single Availability Zone**

○ **To support throughput near or exceeding 20K packets per second (PPS) on the VIF driver** **(Correct)**

○ **To reach speeds up to 2,500 Gbps between EC2 instances**

Explanation

Correct option:

To support throughput near or exceeding 20K packets per second (PPS) on the VIF driver - Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported 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. Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies. There is no additional charge for using enhanced networking.

Consider using enhanced networking for the following scenarios:

1. If your packets-per-second rate reaches its ceiling, consider moving to enhanced networking. If your rate reaches its ceiling, you've likely reached the upper thresholds of the virtual network interface driver.
2. If your throughput is near or exceeding 20K packets per second (PPS) on the VIF driver, it's a best practice to use enhanced networking.

All current generation instance types support enhanced networking, except for T2 instances.

Incorrect options:

To reach speeds up to 2,500 Gbps between EC2 instances - If you need to reach speeds up to 25 Gbps between instances, launch instances in a cluster placement group along with ENA compatible instances. If you need to reach speeds up to 10 Gbps between instances, launch your instances into a cluster placement group with the enhanced networking instance type. This option has been added as a distractor, as it is not possible to support speeds up to 2,500 Gbps between EC2 instances.

To configure multi-attach for an EBS volume that can be attached to a maximum of 16 EC2 instances in a single Availability Zone - An EBS (io1 or io2) volume, when configured with the new Multi-Attach option, can be attached to a maximum of 16 EC2 instances in a single Availability Zone. Additionally, each Nitro-based EC2 instance can support the attachment of multiple Multi-Attach enabled EBS volumes. Multi-Attach capability makes it easier to achieve higher availability for applications that provide write-ordering to maintain storage consistency. You do not need to use enhanced networking to configure this option.

To configure Direct Connect to reach speeds up to 25 Gbps between EC2 instances - AWS Direct Connect is a networking service that provides an alternative to using the internet to connect your on-premises resources to AWS Cloud. In many circumstances, private network connections can reduce costs, increase bandwidth, and provide a more consistent network experience than internet-based connections. You cannot use enhanced networking to configure Direct Connect to reach speeds up to 25 Gbps between EC2 instances.

References:

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

https://aws.amazon.com/premiumsupport/knowledge-center/enable-configure-enhanced-networking/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes-multi.html

Question 55: **Correct**

A multi-national company extensively uses AWS CloudFormation to model and provision its AWS resources. A human error had earlier deleted a critical service from the CloudFormation stack that resulted in business loss. The company is looking at a quick and effective solution to lock the critical resources from any updates or deletes.

As a SysOps Administrator, what will you suggest to address this requirement?

○ **Use parameter constraints to specify the Identities that can update the Stack**

○ **Use Stack policies to protect critical stack resources from unintentional updates** **(Correct)**

○ **Use revision controls to protect critical stack resources from unintentional updates**

○ **Use nested stacks that will retain the configuration in the parent configuration even if the child configuration is lost or cannot be used**

Explanation

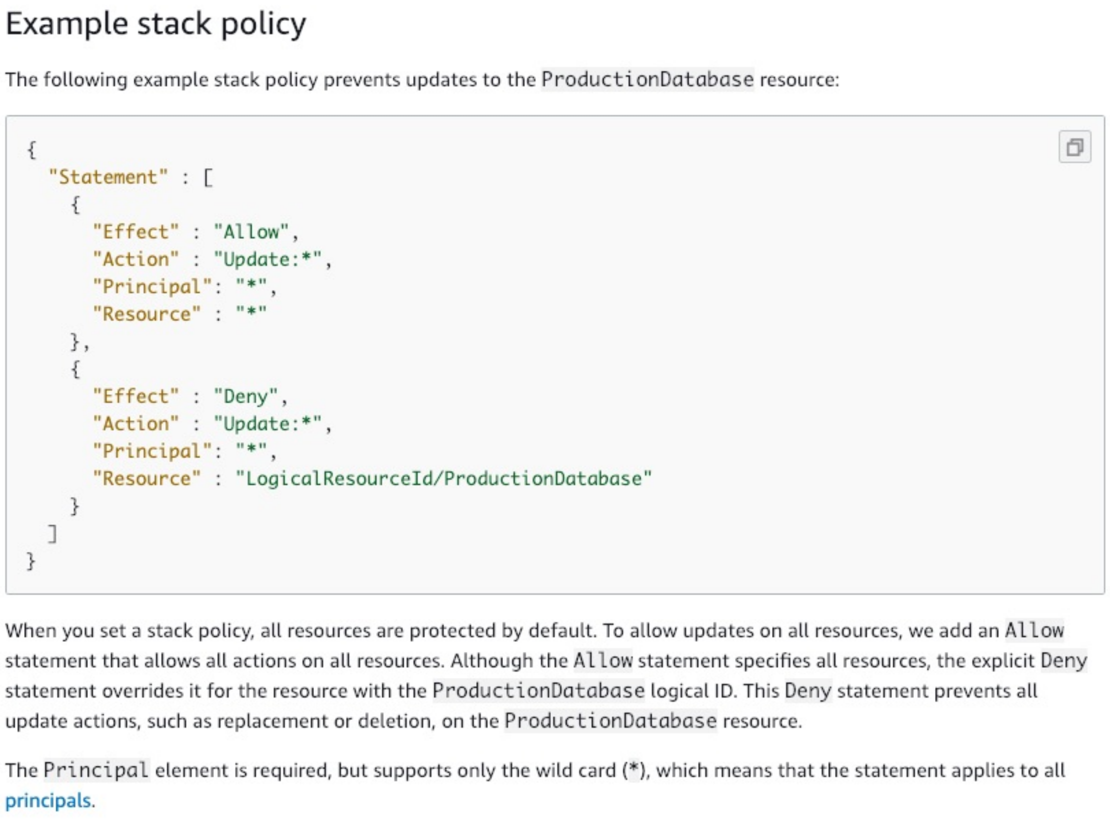
Correct option:

Use Stack policies to protect critical stack resources from unintentional updates

Stack policies help protect critical stack resources from unintentional updates that could cause resources to be interrupted or even replaced. A stack policy is a JSON document that describes what update actions can be performed on designated resources. Specify a stack policy whenever you create a stack that has critical resources.

During a stack update, you must explicitly specify the protected resources that you want to update; otherwise, no changes are made to protected resources.

Example Stack policy:



Example stack policy

The following example stack policy prevents updates to the `ProductionDatabase` resource:

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

When you set a stack policy, all resources are protected by default. To allow updates on all resources, we add an `Allow` statement that allows all actions on all resources. Although the `Allow` statement specifies all resources, the explicit `Deny` statement overrides it for the resource with the `ProductionDatabase` logical ID. This `Deny` statement prevents all update actions, such as replacement or deletion, on the `ProductionDatabase` resource.

The `Principal` element is required, but supports only the wild card (*), which means that the statement applies to all principals.

via

- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/protect-stack-resources.html

Incorrect options:

Use nested stacks that will retain the configuration in the parent configuration even if the child configuration is lost or cannot be used - Nested stacks are stacks that create other stacks. As your infrastructure grows, common patterns can emerge in which you declare the same components in each of your templates. You can separate these common components and create dedicated templates for them. Nested stacks make it easy to manage resources, but it does not protect them from updation.

Use revision controls to protect critical stack resources from unintentional updates - Your stack templates describe the configuration of your AWS resources, such as their property values. To review changes and to keep an accurate history of your resources, use code reviews and revision controls. Although it's a useful feature, it is not relevant for the current scenario.

Use parameter constraints to specify the Identities that can update the Stack - With constraints, you can describe allowed input values so that AWS CloudFormation catches any invalid values before creating a stack. You can set constraints such as a minimum length, maximum length, and allowed patterns. However, you cannot protect resources from deletion.

Reference:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html#nested

Question 56: **Correct**

A media company stores all their articles on Amazon S3 buckets. As a security measure, they have server access logging enabled for all the buckets. The company is looking at a solution that can regularly check if logging is enabled for all the existing buckets and for any new ones they create. If the solution can also automate the remedy, it will be a perfect fit for their requirement.

Which of the following would you suggest to address the given use-case?

○ **Create a Lambda function that will check the logging status of all the S3 buckets and raise an Amazon SNS notification, if a remedy is needed**

○ **Use AWS Config rules to check whether or not an S3 bucket has logging enabled, and carry out the necessary remediation if needed** **(Correct)**

○ **Enable AWS CloudTrail to track the logging information for all the S3 buckets. Currently, AWS does not provide an automatic remediation process, hence, use a Lambda function to rectify any aberrations found during the checks**

○ **Amazon S3 server access logging is checked by AWS Trusted Advisor, as part of the best practices check it performs. Configure a remedy action with Trusted Advisor for all the resources that fails this best practice check**

Explanation

Correct option:

Use AWS Config rules to check whether or not an S3 bucket has logging enabled, and carry out the necessary remediation if needed

AWS Config keeps track of the configuration of your AWS resources and their relationships to other resources. It can also evaluate those AWS resources for compliance. This service uses rules that can be configured to evaluate AWS resources against desired configurations.

For example, there are AWS Config rules that check whether or not your Amazon S3 buckets have logging enabled or your IAM users have an MFA device enabled. AWS Config rules use AWS Lambda functions to perform the compliance evaluations, and the Lambda functions return the compliance status of the evaluated resources as compliant or noncompliant. The non-compliant resources are remediated using the remediation action associated with the AWS Config rule. With the Auto-Remediation feature of AWS Config rules, the remediation action can be executed automatically when a resource is found non-compliant.

AWS Config Auto Remediation feature has auto remediate feature for any non-compliant S3 buckets using the following AWS Config rules:

s3-bucket-logging-enabled s3-bucket-server-side-encryption-enabled s3-bucket-public-read-prohibited s3-bucket-public-write-prohibited

These AWS Config rules act as controls to prevent any non-compliant S3 activities.

Incorrect options:

Create a Lambda function that will check the logging status of all the S3 buckets and raise an Amazon SNS notification, if a remedy is needed - AWS Lambda cannot poll by itself and needs a polling mechanism or a service to invoke it. Any logic written to self invoke in the same Lambda function will result in Lambda taking up all the resources available and running continuously, which will also become an expensive solution.

Enable AWS CloudTrail to track the logging information for all the S3 buckets. Currently, AWS does not provide an automatic remediation process, hence, use a Lambda function to rectify any aberrations found during the checks - As discussed above, remediation action is possible with AWS Config and hence is the right solution here.

Amazon S3 server access logging is checked by AWS Trusted Advisor, as part of the best practices check it performs. Configure a remedy action with Trusted Advisor for all the resources that fails this best practice check - AWS Trusted Advisor checks the configuration of Amazon Simple Storage Service (Amazon S3) buckets that have server access logging enabled. It recommends action after these checks but cannot automate a remedial action. Hence, Trusted advisor is not an optimal solution for the current scenario.

Reference:

https://aws.amazon.com/blogs/mt/aws-config-auto-remediation-s3-compliance/

Question 57: **Incorrect**

As a SysOps Administrator, you maintain the development account of a large team that comprises of both developers and testers. The Development account has two IAM groups: Developers and Testers. Users in both groups have permission to work in the development account and access resources there. From time to time, a developer must update the live S3 Bucket in the production account.

How will you configure the permissions for developers to access the production environment?

○ **Create a Role in development account, that defines the production account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Then, modify the IAM group policy in development account, so that testers are denied access to the newly created role. Developers can use the newly created role to access the live S3 buckets in production environment** (Incorrect)

○ **Create a Role in production account, that defines the development account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Then, modify the IAM group policy in development account, so that testers are denied access to the newly created role. Developers can use the newly created role to access the live S3 buckets in production environment** (Correct)

○ **Use Inline policies to be sure that the permissions in a policy are not inadvertently assigned to an identity other than the one they're intended for**

○ **Create a Role in Production account, that defines the Development account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Developers can use the newly created role to access the live S3 buckets in production environment**

Explanation

Correct option:

Create a Role in production account, that defines the Development account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Then, modify the IAM group policy in development account, so that testers are denied access to the newly created role. Developers can use the newly created role to access the live S3 buckets in production environment -

First, you use the AWS Management Console to establish trust between the production account and the development account. You start by creating an IAM role. When you create the role, you define the development account as a trusted entity and specify a permissions policy that allows trusted users to update the production bucket.

You need to then modify the IAM group policy so that Testers are explicitly denied access to the created role.

Finally, as a developer, you use the created role to update the bucket in the Production account.

Incorrect options:

Create a Role in development account, that defines the production account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Then, modify the IAM group policy in development account, so that testers are denied access to the newly created role. Developers can use the newly created role to access the live S3 buckets in production environment - Role has to be created in production account since the resource to be accessed is in this account.

Use Inline policies to be sure that the permissions in a policy are not inadvertently assigned to an identity other than the one they're intended for - An inline policy is a policy that's embedded in an IAM identity (a user, group, or role). That is, the policy is an inherent part of the identity. You can create a policy and embed it in an identity, either when you create the identity or later.

Inline policies are useful if you want to maintain a strict one-to-one relationship between a policy and the identity that it's applied to. For example, you want to be sure that the permissions in a policy are not inadvertently assigned to an identity other than the one they're intended for. When you use an inline policy, the permissions in the policy cannot be inadvertently attached to the wrong identity.

Create a Role in Production account, that defines the Development account as a trusted entity and specify a permissions policy that allows trusted users to update the bucket. Developers can use the newly created role to access the live S3 buckets in production environment - This option does not deny access to Testers, so it is not correct.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#customer-managed-policies

Question 58: **Correct**

A company initially used a manual process to create and manage different IAM roles needed for the organization. As the company expanded and lines of business grew, different AWS accounts were created to manage the AWS resources as well as the users. The manual process has resulted in errors with IAM roles getting created with insufficient permissions. The company is looking at automating the process of creating and managing the necessary IAM roles for multiple AWS accounts. The company already uses AWS Organizations to manage multiple AWS accounts.

As a SysOps Administrator, can you suggest an effective way to automate this process?

○ **Create CloudFormation templates and reuse them to create necessary IAM roles in each of the AWS accounts**

○ **Use AWS Resource Access Manager that integrates with AWS Organizations to deploy and manage shared resources across AWS accounts**

○ **Use AWS Directory Service with AWS Organizations to automatically associate necessary IAM roles with the Microsoft Active Directory users**

○ **Use CloudFormation StackSets with AWS Organizations to deploy and manage IAM roles to multiple AWS accounts simultaneously** **(Correct)**

Explanation

Correct option:

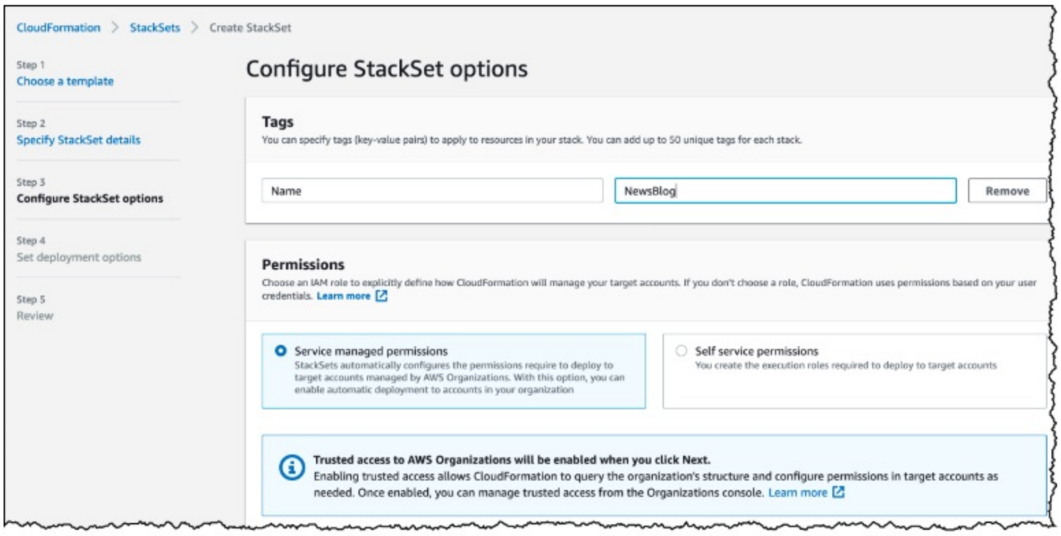
Use CloudFormation StackSets with AWS Organizations to deploy and manage IAM roles to multiple AWS accounts simultaneously

CloudFormation StackSets allow you to roll out CloudFormation stacks over multiple AWS accounts and in multiple Regions with just a couple of clicks. When AWS launched StackSets, grouping accounts was primarily for billing purposes. Since the launch of AWS Organizations, you can centrally manage multiple AWS accounts across diverse business needs including billing, access control, compliance, security and resource sharing.

You can now centrally orchestrate any AWS CloudFormation enabled service across multiple AWS accounts and regions. For example, you can deploy your centralized AWS Identity and Access Management (IAM) roles, provision Amazon Elastic Compute Cloud (EC2) instances or AWS Lambda functions across AWS Regions and accounts in your organization. CloudFormation StackSets simplify the configuration of cross-accounts permissions and allow for automatic creation and deletion of resources when accounts are joining or are removed from your Organization.

You can get started by enabling data sharing between CloudFormation and Organizations from the StackSets console. Once done, you will be able to use StackSets in the Organizations master account to deploy stacks to all accounts in your organization or in specific organizational units (OUs). A new service managed permission model is available with these StackSets. Choosing Service managed permissions allows StackSets to automatically configure the necessary IAM permissions required to deploy your stack to the accounts in your organization.

How to use AWS CloudFormation StackSets for Multiple Accounts in an AWS Organization:



CloudFormation > StackSets > Create StackSet

Step 1
Choose a template

Step 2
Specify StackSet details

Step 3
Configure StackSet options

Step 4
Set deployment options

Step 5
Review

Configure StackSet options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack.

Name | NewsBlog | Remove

Permissions

Choose an IAM role to explicitly define how CloudFormation will manage your target accounts. If you don't choose a role, CloudFormation uses permissions based on your user credentials. Learn more

● Service managed permissions
StackSets automatically configures the permissions require to deploy to target accounts managed by AWS Organizations. With this option, you can enable automatic deployment to accounts in your organization

○ Self service permissions
You create the execution roles required to deploy to target accounts

Trusted access to AWS Organizations will be enabled when you click Next.
Enabling trusted access allows CloudFormation to query the organization's structure and configure permissions in target accounts as needed. Once enabled, you can manage trusted access from the Organizations console. Learn more

via

- https://aws.amazon.com/blogs/aws/new-use-aws-cloudformation-stacksets-for-multiple-accounts-in-an-aws-organization/

Incorrect options:

Create CloudFormation templates and reuse them to create necessary IAM roles in each of the AWS accounts - CloudFormation templates can ease the current manual process that the company is using. However, it's not a completely automated process that the company needs.

Use AWS Directory Service with AWS Organizations to automatically associate necessary IAM roles with the Microsoft Active Directory users - AWS Directory Service for Microsoft Active Directory, or AWS Managed Microsoft AD, lets you run Microsoft Active Directory (AD) as a managed service. AWS Directory Service makes it easy to set up and run directories in the AWS Cloud or connect your AWS resources with an existing on-premises Microsoft Active Directory. It is not meant for the automatic creation of IAM roles across AWS accounts.

Use AWS Resource Access Manager that integrates with AWS Organizations to deploy and manage shared resources across AWS accounts - AWS Resource Access Manager (AWS RAM) enables you to share specified AWS resources that you own with other AWS accounts. It's a centralized service that provides a consistent experience for sharing different types of AWS resources across multiple accounts. This service enables you to share resources across AWS accounts. It's not meant for re-creating the same resource definitions in different AWS accounts.

References:

https://aws.amazon.com/blogs/aws/new-use-aws-cloudformation-stacksets-for-multiple-accounts-in-an-aws-organization/

https://docs.aws.amazon.com/organizations/latest/userguide/services-that-can-integrate-ram.html

Question 59: **Incorrect**

A video streaming app uses Amazon Kinesis Data Streams for streaming data. The systems administration team needs to be informed of the shard capacity when it is reaching its limits.

How will you configure this requirement?

○ **Configure Amazon CloudTrail to generate logs for the service limits. CloudTrail and CloudWatch are integrated and hence alarm can be generated for customized service checks**

○ **Configure Amazon CloudWatch Events to pick data from Amazon Inspector**

○ **Monitor Trusted Advisor service check results with Amazon CloudWatch Events** **(Correct)**

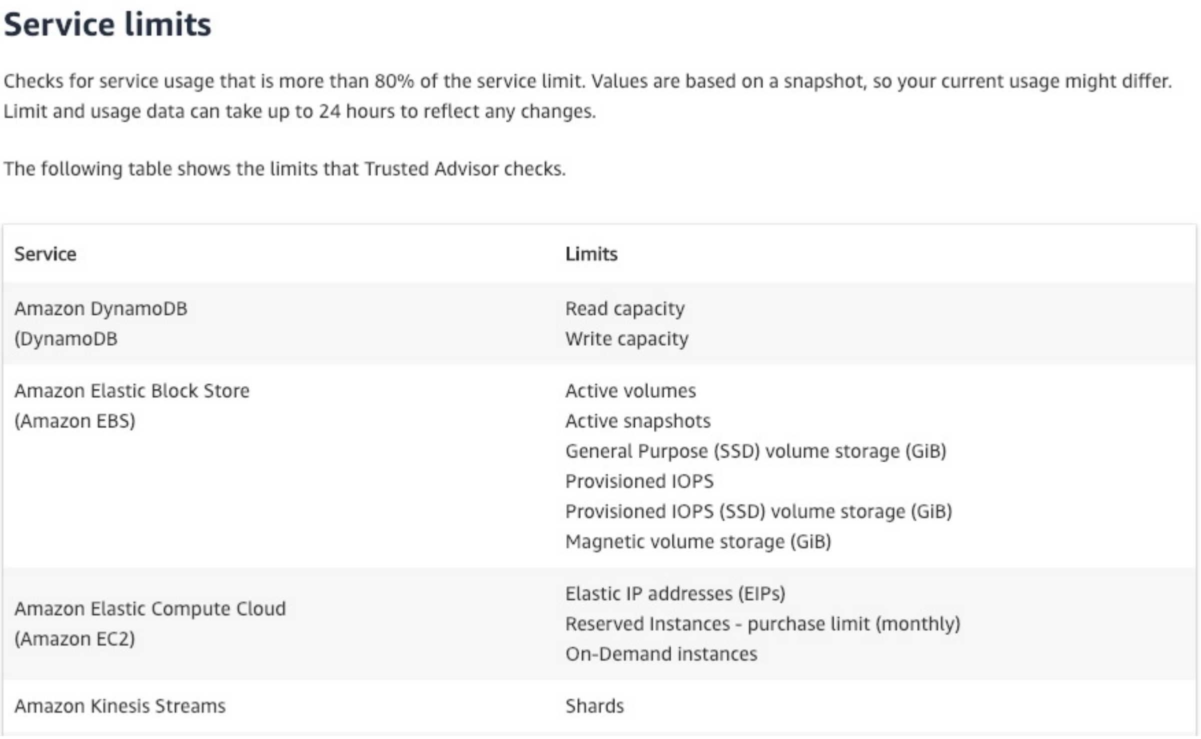
○ **Use CloudWatch ServiceLens to monitor data on service limits of various AWS services** **(Incorrect)**

Explanation

Correct option:

Monitor Trusted Advisor service check results with Amazon CloudWatch Events - AWS Trusted Advisor checks for service usage that is more than 80% of the service limit.

A partial list of Trusted Advisor service limit checks:



Service limits

Checks for service usage that is more than 80% of the service limit. Values are based on a snapshot, so your current usage might differ. Limit and usage data can take up to 24 hours to reflect any changes.

The following table shows the limits that Trusted Advisor checks.

Service	Limits
Amazon DynamoDB (DynamoDB	Read capacity Write capacity
Amazon Elastic Block Store (Amazon EBS)	Active volumes Active snapshots General Purpose (SSD) volume storage (GiB) Provisioned IOPS Provisioned IOPS (SSD) volume storage (GiB) Magnetic volume storage (GiB)
Amazon Elastic Compute Cloud (Amazon EC2)	Elastic IP addresses (EIPs) Reserved Instances - purchase limit (monthly) On-Demand instances
Amazon Kinesis Streams	Shards

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

You can use Amazon CloudWatch Events to detect and react to changes in the status of Trusted Advisor checks. Then, based on the rules that you create, CloudWatch Events invokes one or more target actions when a status check changes to the value you specify in a rule. Depending on the type of status change, you might want to send notifications, capture status information, take corrective action, initiate events, or take other actions.

Incorrect options:

Configure Amazon CloudWatch Events to pick data from 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. Not the right service for the given requirement.

Use CloudWatch ServiceLens to monitor data on service limits of various AWS services - CloudWatch ServiceLens enhances the observability of your services and applications by enabling you to integrate traces, metrics, logs, and alarms into one place. So, ServiceLens can be used once we define the alarms in CloudWatch, not without it.

Configure Amazon CloudTrail to generate logs for the service limits. CloudTrail and CloudWatch are integrated and hence alarm can be generated for customized service checks - 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 however, does not monitor service limits.

References:

https://docs.aws.amazon.com/awssupport/latest/user/cloudwatch-events-ta.html

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ServiceLens.html https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/

Question 60: **Correct**

An application runs on a fleet of Amazon EC2 instances running behind an Application Load Balancer. An Auto Scaling Group (ASG) helps keep the application available and flexible to traffic changes. The EC2 instances need to connect to Amazon RDS instances for fetching data. EC2 Instances also need internet access to be able to download the patches needed for their software. To meet the security guidelines of the company - the Load Balancer, Auto Scaling Group with the EC2 instances and RDS - are all placed into different subnets of the VPC.

Which of the following represents the best configuration to help connect the EC2 instances to the internet?

○ **Create and attach an Egress-only Internet Gateway to the VPC and then update the route table of the instance subnet to route internet traffic via the Egress-only Internet Gateway**

○ **Create a carrier gateway and attach the carrier gateway to your VPC. You can then connect the subnets you wish to route to the carrier gateway**

○ **Create and attach an Internet Gateway to the VPC. Update the route table of the subnet that hosts the EC2 instances, to route internet traffic via the Internet Gateway** **(Correct)**

○ **Configure an Elastic network interface for all the instances that need to communicate with the internet. Attach this Elastic network interface to the public subnet of the VPC to route internet traffic**

Explanation

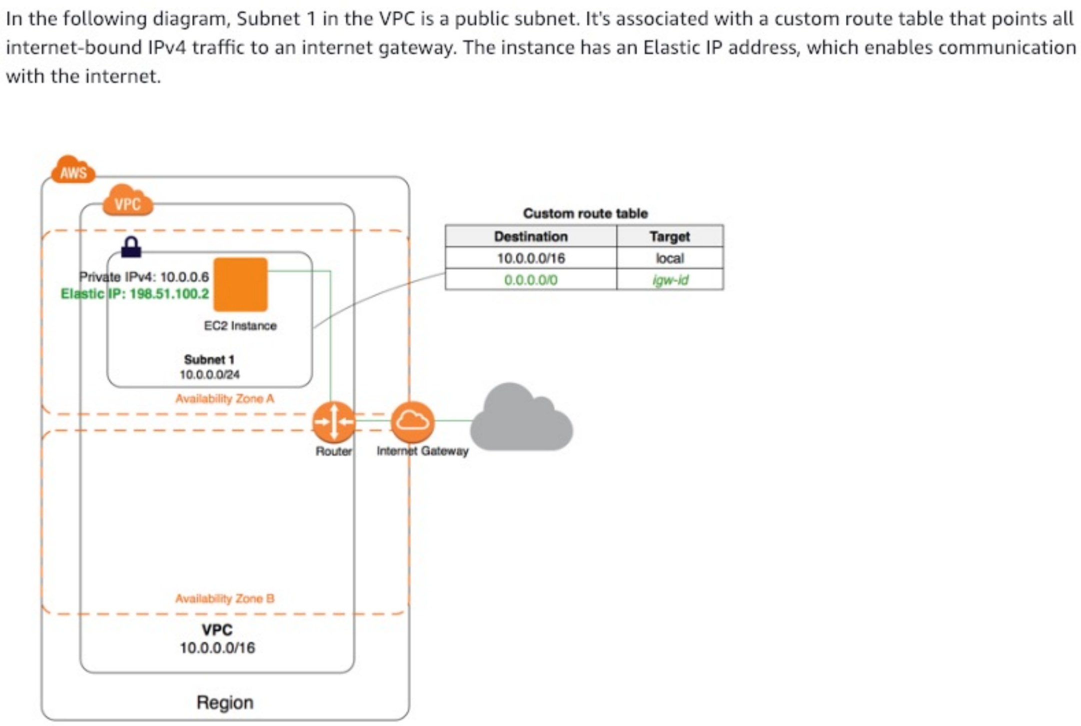
Correct option:

Create and attach an Internet Gateway to the VPC. Update the route table of the subnet that hosts the EC2 instances, to route internet traffic via the Internet Gateway

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses. An internet gateway supports IPv4 and IPv6 traffic. It does not cause availability risks or bandwidth constraints on your network traffic. There's no additional charge for having an internet gateway in your account.

Reference diagram to configure an Internet Gateway on a VPC:



In the following diagram, Subnet 1 in the VPC is a public subnet. It's associated with a custom route table that points all internet-bound IPv4 traffic to an internet gateway. The instance has an Elastic IP address, which enables communication with the internet.

Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

via

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

Incorrect options:

Create and attach an Egress-only Internet Gateway to the VPC and then update the route table of the instance subnet to route internet traffic via the Egress-only Internet Gateway - An Egress-only internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the internet, and prevents the internet from initiating an IPv6 connection with your instances. An egress-only internet gateway is for use with IPv6 traffic only.

Configure an Elastic network interface for all the instances that need to communicate with the internet. Attach this Elastic network interface to the public subnet of the VPC to route internet traffic - An elastic network interface is a virtual network interface that can include the following attributes: a primary private IPv4 address, one or more secondary private IPv4 addresses, one Elastic IP address per private IPv4 address, one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance, one or more IPv6 addresses, one or more security groups, a MAC address , a source/destination check flag, a description.

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

This option is a distractor, as there is no such thing as attaching a Elastic network interface to the public subnet of the VPC.

Create a carrier gateway and attach the carrier gateway to your VPC. You can then connect the subnets you wish to route to the carrier gateway - A carrier gateway serves two purposes. It allows inbound traffic from a carrier network in a specific location, and it allows outbound traffic to the carrier network and the internet. There is no inbound connection configuration from the internet to a Wavelength Zone through the carrier gateway. Carrier gateways are only available for VPCs that contain subnets in a Wavelength Zone.

References:

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

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

Question 61: **Incorrect**

Multiple teams of an e-commerce company use the same AWS CloudFormation template to create stacks of resources needed by them. For the next deployment, the teams need to update the stacks and have been testing the changes through change sets. However, the teams suddenly realized that all their change sets have been lost. Unable to figure out the error they have approached you.

As a SysOps Administrator, how will you identify the error and suggest a way to fix the issue?

○ **An invalid change set was executed and this resulted in all stacks and change sets getting deleted**

○ **CloudFormation had issued a rollback on the change sets while validating them and deleted all the invalid sets** (Incorrect)

○ **The change set while being validated, surpassed the account limit of some AWS resource. Since the stacks cannot be updated when the account limit is reached, the change sets have been deleted by CloudFormation**

○ **A change set was successfully executed and this resulted in rest of the change sets being deleted by CloudFormation** (Correct)

Explanation

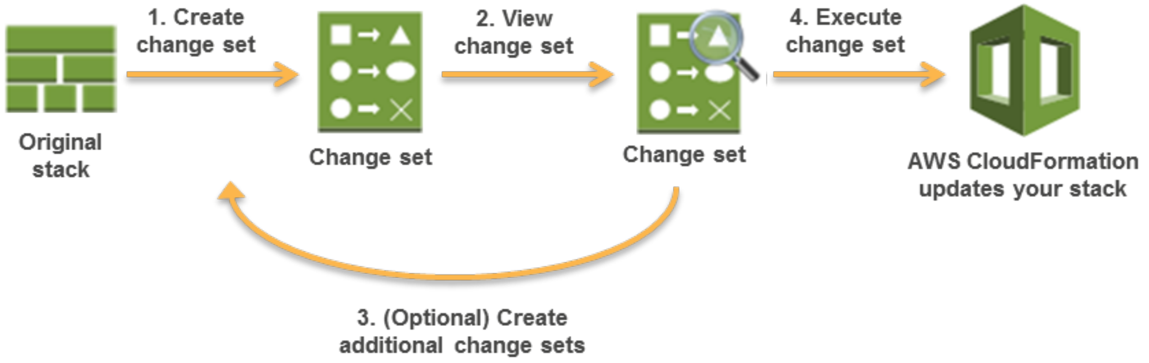
Correct option:

A change set was successfully executed and this resulted in rest of the change sets being deleted by CloudFormation

Change sets allow you to preview how proposed changes to a stack might impact your existing resources, for example, whether your changes will delete or replace any critical resources, AWS CloudFormation makes the changes to your stack only when you decide to execute the change set, allowing you to decide whether to proceed with your proposed changes or explore other changes by creating another change set. You can create and manage change sets using the AWS CloudFormation console, AWS CLI, or AWS CloudFormation API.

After you execute a change, AWS CloudFormation removes all change sets that are associated with the stack because they aren't applicable to the updated stack.

How to use change sets to update a stack:



via

- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html

Incorrect options:

An invalid change set was executed and this resulted in all stacks and change sets getting deleted - This option has been added as a distractor. An invalid change set won't result in any resource changes as it won't go through for provisioning.

The change set while being validated, surpassed the account limit of some AWS resource. Since the stacks cannot be updated when the account limit is reached, the change sets have been deleted by CloudFormation - Change sets don't indicate whether AWS CloudFormation will successfully update a stack. For example, a change set doesn't check if you will surpass an account limit, if you're updating a resource that doesn't support updates, or if you have insufficient permissions to modify a resource, all of which can cause a stack update to fail. If an update fails, AWS CloudFormation attempts to roll back your resources to their original state.

CloudFormation had issued a rollback on the change sets while validating them and deleted all the invalid sets - There is no rollback for change sets, since there is no real change. When they are applied on a stack and stack fails, the stack is rolled back to its previous state.

Reference:

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html

Question 62: **Incorrect**

Consider this scenario - the primary instance of an Amazon Aurora cluster is unavailable because of an outage that has affected an entire AZ. The primary instance and all the reader instances are in the same AZ.

As a SysOps Administrator, what action will you take to get the database online?

○ **Aurora promotes an existing replica in another AZ to a new primary instance, so nothing needs to be done** **(Incorrect)**

○ **For a cluster using single-master replication, Aurora can create up to 15 read-only Aurora Replicas to serve requests from users**

○ **Aurora automatically creates a new primary instance in the same AZ**

○ **You must manually create one or more new DB instances in another AZ** **(Correct)**

Explanation

Correct option:

You must manually create one or more new DB instances in another AZ

Suppose that the primary instance in your cluster is unavailable because of an outage that affects an entire AZ. In this case, the way to bring a new primary instance online depends on whether your cluster uses a multi-AZ configuration. If the cluster contains any reader instances in other AZs, Aurora uses the failover mechanism to promote one of those reader instances to be the new primary instance. If your provisioned cluster only contains a single DB instance, or if the primary instance and all reader instances are in the same AZ, you must manually create one or more new DB instances in another AZ.

Incorrect options:

Aurora promotes an existing replica in another AZ to a new primary instance - The use case states that the primary instance and all the reader instances are in the same AZ. So, this is not possible.

Aurora automatically creates a new primary instance in the same AZ - If the primary instance in a DB cluster using single-master replication fails, Aurora automatically fails over to a new primary instance in one of two ways:

1. By promoting an existing Aurora Replica to the new primary instance
2. By creating a new primary instance

But, in this use case, the AZ itself has failed. So, creating a new primary in the same AZ is not possible.

For a cluster using single-master replication, Aurora can create up to 15 read-only Aurora Replicas to serve requests from users - Generally, an Aurora DB cluster can contain up to 15 Aurora Replicas. The Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region. But, this use case is a single AZ deployment with failure at the AZ level. So, this solution is not possible.

Reference:

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

Question 63: **Correct**

A data analytics company has its server infrastructure built on Amazon EC2 instances fronted with Elastic Load Balancers (ELBs). The ELBs are maintained in two AZs with each ELB having two EC2 instances registered with it. Both the instances in one AZ have been recorded as unhealthy.

What is the status of traffic that flows to the ELB connected to unhealthy instances?

○ **The Load Balancer will display an `unhealthy' status and will not accept any incoming requests**

○ **The Load Balancer routes requests to the unhealthy targets** **(Correct)**

○ **HTTP 403: Forbidden will be returned**

○ **HTTP 503: Service unavailable will be received as response**

Explanation

Correct option:

The Load Balancer routes requests to the unhealthy targets - If there is at least one healthy target in a target group, the load balancer routes requests only to the healthy targets. If a target group contains only unhealthy targets, the load balancer routes requests to the unhealthy targets. Hence, it is advised to configure an Auto Scaling Group, if the instances are hosting a business-critical application.

Incorrect options:

HTTP 503: Service unavailable will be received as response - 503 error is returned if the target groups for the load balancer have no registered targets.

The Load Balancer will display an `unhealthy' status and will not accept any incoming requests - This is a made-up option, given only as a distractor.

HTTP 403: Forbidden will be returned - 403 error is returned if you configured an AWS WAF web access control list (web ACL) to monitor requests to your Application Load Balancer and it blocked the request.

Reference:

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

Question 64: **Incorrect**

A development team has configured its AWS VPC with one public and one private subnet. The public subnet has an Amazon EC2 instance that hosts the application. The private subnet has the RDS database that the application needs to communicate with.

Which of the following would you identify as the correct way to configure a solution for the given requirement?

○ **Elastic IP can be configured to initiate communication between private and public subnets** **(Incorrect)**

○ **Create a Security Group that allows connection from different subnets inside a VPC**

○ **Configure a VPC peering for enabling communication between the subnets**

○ **Subnets inside a VPC can communicate with each other without the need for any further configuration. Hence, no additional configurations are needed** **(Correct)**

Explanation

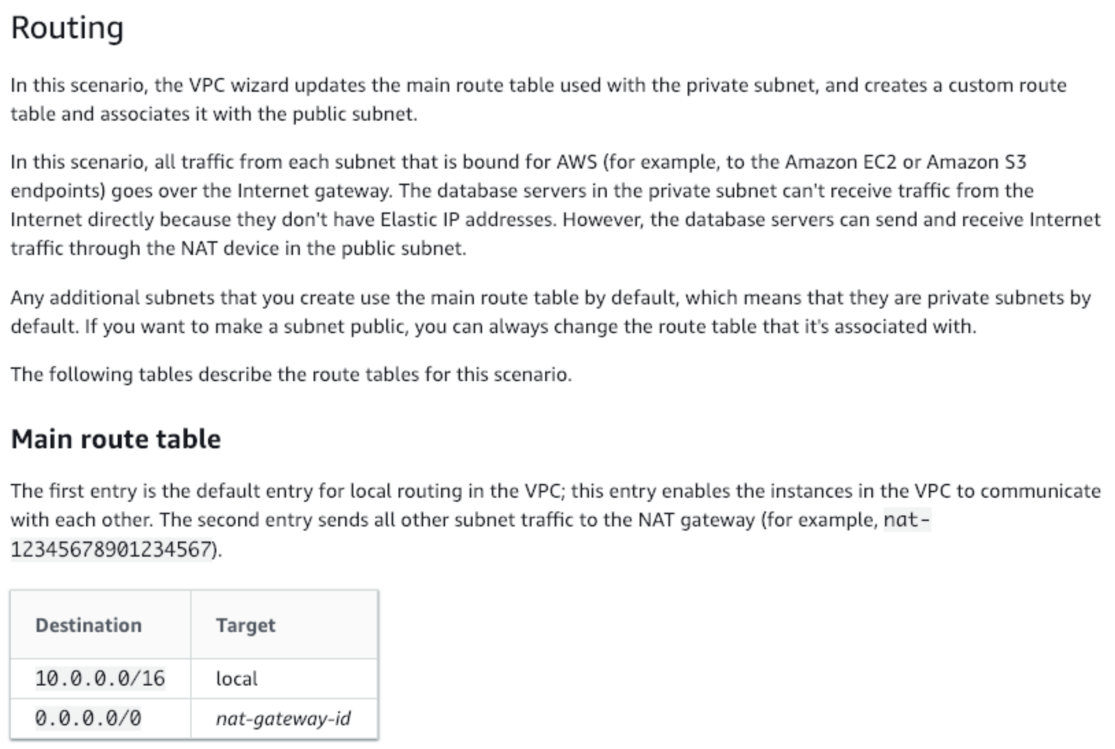
Correct option:

Subnets inside a VPC can communicate with each other without the need for any further configuration. Hence, no additional configurations are needed - Subnets inside a VPC can communicate with each other without any additional configurations.

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.

A route table contains a set of rules, called routes, that are used to determine where network traffic from your VPC is directed. You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table.

The first entry in the Main route table is the default entry for local routing in the VPC; this entry enables the instances (potentially belonging to different subnets) in the VPC to communicate with each other.



Routing

In this scenario, the VPC wizard updates the main route table used with the private subnet, and creates a custom route table and associates it with the public subnet.

In this scenario, all traffic from each subnet that is bound for AWS (for example, to the Amazon EC2 or Amazon S3 endpoints) goes over the Internet gateway. The database servers in the private subnet can't receive traffic from the Internet directly because they don't have Elastic IP addresses. However, the database servers can send and receive Internet traffic through the NAT device in the public subnet.

Any additional subnets that you create use the main route table by default, which means that they are private subnets by default. If you want to make a subnet public, you can always change the route table that it's associated with.

The following tables describe the route tables for this scenario.

Main route table

The first entry is the default entry for local routing in the VPC; this entry enables the instances in the VPC to communicate with each other. The second entry sends all other subnet traffic to the NAT gateway (for example, `nat-12345678901234567`).

Destination	Target
`10.0.0.0/16`	local
`0.0.0.0/0`	*nat-gateway-id*

via - https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html

Incorrect options:

Elastic IP can be configured to initiate communication between private and public subnets - An Elastic IP address is a reserved public IP address that you can assign to any EC2 instance in a particular region until you choose to release it. Elastic IP is not needed for resources to talk across subnets in the same VPC.

Configure a VPC peering for enabling communication between the subnets - 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. It is not needed for resources inside the same VPC.

Create a Security Group that allows connection from different subnets inside a VPC - A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html

https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html#what-is-route-tables

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

Question 65: **Incorrect**

A development team has written configurable scripts that need to be run every day to monitor the business endpoints and APIs. The team wants to integrate these scripts with Amazon CloudWatch service to help in overall monitoring and analysis.

What is the right way of configuring this requirement?

○ **Use CloudWatch ServiceLens to integrate the custom script into CloudWatch system for generating metrics and logs** **(Incorrect)**

○ **CloudWatch Dashboard settings can be used to integrate the user-written scripts into Alarms generated and managed by CloudWatch**

○ **Use CloudWatch Synthetics to create canaries which create CloudWatch metrics to track and monitor the services** **(Correct)**

○ **Configure a CloudWatch Composite Alarm and integrate the configurable script, written by the team, with the CloudWatch logs**

Explanation

Correct option:

Use CloudWatch Synthetics to create canaries, which create CloudWatch metrics to track and monitor the services - You can use Amazon CloudWatch Synthetics to create canaries, configurable scripts that run on a schedule, to monitor your endpoints and APIs. Canaries follow the same routes and perform the same actions as a customer, which makes it possible for you to continually verify your customer experience even when you don't have any customer traffic on your applications. By using canaries, you can discover issues before your customers do.

Canaries are Node.js scripts. They create Lambda functions in your account that use Node.js as a framework. Canaries work over both HTTP and HTTPS protocols.

UI canaries offer programmatic access to a headless Google Chrome Browser via Puppeteer. For more information about Puppeteer, see Puppeteer.

Canaries check the availability and latency of your endpoints and can store load time data and screenshots of the UI. They monitor your REST APIs, URLs, and website content, and they can check for unauthorized changes from phishing, code injection and cross-site scripting.

You can run a canary once or on a regular schedule. Scheduled canaries can run 24 hours a day, as often as once per minute.

Incorrect options:

Configure a CloudWatch Composite Alarm and integrate the configurable script, written by the team, with the CloudWatch logs - 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. Not the right choice for the current scenario.

CloudWatch Dashboard settings can be used to integrate the user-written scripts into Alarms generated and managed by CloudWatch - This is a made-up option, given only as a distractor.

Use CloudWatch ServiceLens to integrate the custom script into CloudWatch system for generating metrics and logs - CloudWatch ServiceLens enhances the observability of your services and applications by enabling you to integrate traces, metrics, logs, and alarms into one place. ServiceLens integrates CloudWatch with AWS X-Ray to provide an end-to-end view of your application to help you more efficiently pinpoint performance bottlenecks and identify impacted users. A very useful service, but not for our current requirement.