SIMPLE STORAGE SERVICE (S3) - QUIZ

Attempt	6
Marks Obtained	23 / 25
Your score is	92%
Completed on	Monday , 21 January 2019 , 04:07 PM
Time Taken	00 H 02 M 47 S
Result	Pass

Domains / Topics wise Quiz Performance Report

S.No.	Topic	Total Questions	Correct	Incorrect	Unattempted
1	Other	25	23	2	0

25 Questions	**23** Correct	**2** Incorrect	**0** Unattempted

Show Answers

QUESTION 1 CORRECT

You have created an S3 bucket in us-east-1 region by not changing default "configure options" and "permissions". Which of the following options are incorrect in terms of default settings?(choose 2 options)

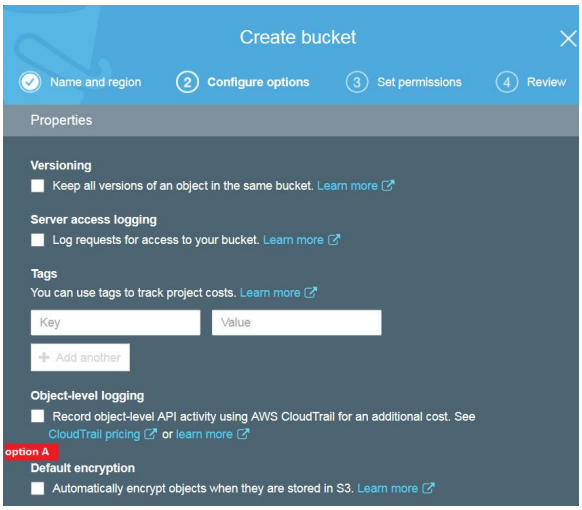
- [ ] **A.** Encryption is disabled.
- [x] **B.** Transfer Acceleration is enabled. ✔
- [ ] **C.** No bucket policy exists.
- [x] **D.** Versioning is enabled. ✔

Explanation :

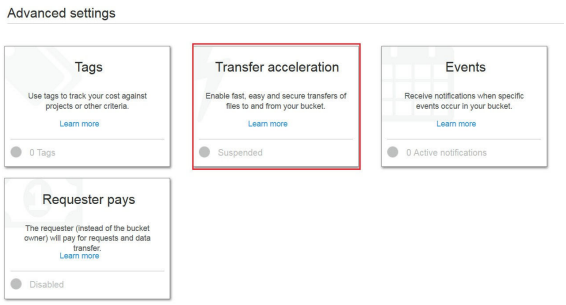
Answer: B, D

When creating an S3 bucket, you can change the default configuration according to your requirements or leave the default options and continue to create the bucket. You can always change the configuration after you created the bucket.

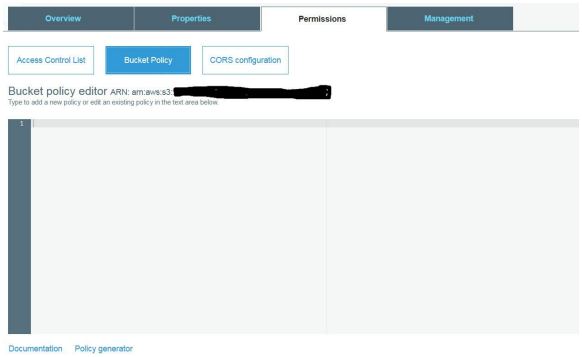
For option A, Default encryption is not enabled.



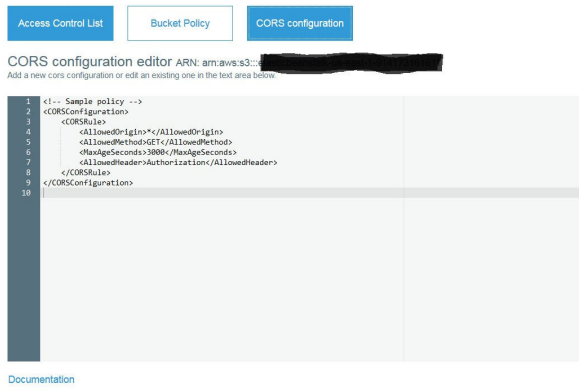
For option B, Transfer Acceleration is suspended by default.



For Option C, bucket policy does not exist by default. We can restrict bucket access through bucket policy.



For option D, By default Versioning is Disabled.



Ask our Experts

QUESTION 2 CORRECT

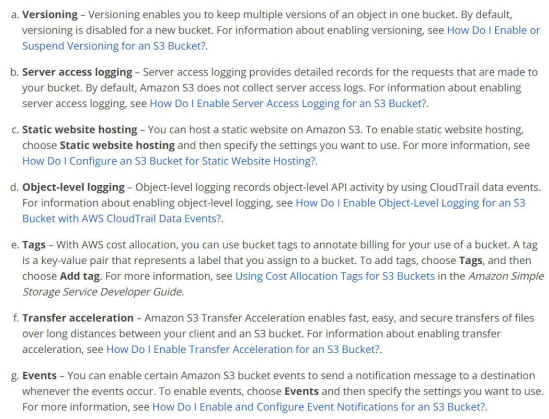
Which of the following are S3 bucket properties?(Choose 2 options)

- [x] **A.** Server access logging ✔
- [x] **B.** Object level logging ✔
- [ ] **C.** Storage class
- [ ] **D.** Metadata

Explanation :

Answer: A, B

Following are S3 bucket properties.



a. **Versioning** – Versioning enables you to keep multiple versions of an object in one bucket. By default, versioning is disabled for a new bucket. For information about enabling versioning, see How Do I Enable or Suspend Versioning for an S3 Bucket?.

b. **Server access logging** – Server access logging provides detailed records for the requests that are made to your bucket. By default, Amazon S3 does not collect server access logs. For information about enabling server access logging, see How Do I Enable Server Access Logging for an S3 Bucket?.

c. **Static website hosting** – You can host a static website on Amazon S3. To enable static website hosting, choose **Static website hosting** and then specify the settings you want to use. For more information, see How Do I Configure an S3 Bucket for Static Website Hosting?.

d. **Object-level logging** – Object-level logging records object-level API activity by using CloudTrail data events. For information about enabling object-level logging, see How Do I Enable Object-Level Logging for an S3 Bucket with AWS CloudTrail Data Events?.

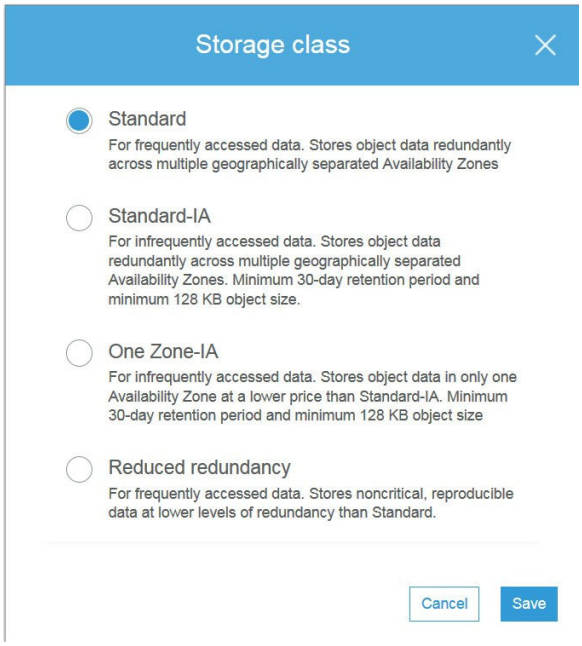
e. **Tags** – With AWS cost allocation, you can use bucket tags to annotate billing for your use of a bucket. A tag is a key-value pair that represents a label that you assign to a bucket. To add tags, choose **Tags**, and then choose **Add tag**. For more information, see Using Cost Allocation Tags for S3 Buckets in the *Amazon Simple Storage Service Developer Guide*.

f. **Transfer acceleration** – Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. For information about enabling transfer acceleration, see How Do I Enable Transfer Acceleration for an S3 Bucket?.

g. **Events** – You can enable certain Amazon S3 bucket events to send a notification message to a destination whenever the events occur. To enable events, choose **Events** and then specify the settings you want to use. For more information, see How Do I Enable and Configure Event Notifications for an S3 Bucket?.

- https://docs.aws.amazon.com/AmazonS3/latest/user-guide/view-bucket-properties.html (https://docs.aws.amazon.com/AmazonS3/latest/user-guide/view-bucket-properties.html)

Option C, Storage class property is at object level, not at bucket level. Following are different storage classes.



For more information on storage classes, refer documentation here.

- https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html (https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html)

For option D, metadata is at object level property, not bucket level.For detailed information on object metadata, refer documentation here. https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingMetadata.html#object-metadata (https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingMetadata.html#object-metadata)

Ask our Experts

QUESTION 3 CORRECT

You have created an S3 bucket in us-east-1 region with default configuration. You are located in Asia and deleted an object in the bucket using AWS CLI. However, when you tried to list the objects in the bucket, you still see the object you deleted. You are even able to download the object. What could have caused this behaviour?

- A. Cross region deletes are not supported by AWS
- B. AWS provides eventual consistency for DELETES. ✔
- C. AWS keeps copy of deleted object for 7 days in STANDARD storage.
- D. AWS provides strong consistency for DELETES.

Explanation :

Answer: B

Amazon S3 offers eventual consistency for overwrite PUTS and DELETES in all regions.

- A process deletes an existing object and immediately lists keys within its bucket. Until the deletion is fully propagated, Amazon S3 might list the deleted object.

https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#CoreConcepts (https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#CoreConcepts) and refer to "Amazon S3 Data Consistency Model"

For option A, you can perform DELETE operation from Console, CLI, programmatically from any region as long as you have access to perform.

For option C, AWS S3 deletes any object for which DELETE request is made from an authorized

IAM entity.

It does not keep a copy unless you have versioning enabled and you have multiple versions of the deleted object.

The DELETE operation removes the null version (if there is one) of an object and inserts a delete marker, which becomes the current version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETE.html (https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETE.html)
In this case, bucket is created with default configuration which has versioning disabled.
For option D, AWS does not provide strong consistency for DELETES.

Ask our Experts

QUESTION 4 CORRECT

Your organization is planning to upload large number of files to AWS cloud. These files need to be immediately available for download across different geographical regions right after the upload is complete. They consulted you to check if S3 is a suitable solution for the use case. What do you suggest?

- **A.** S3 is not suitable for immediate downloads because new AWS provides eventual consistency for new objects.
- **B.** S3 is suitable for immediate downloads because AWS provides read-after-write consistency for new objects. ✔
- **C.** EFS is suitable for immediate downloads because AWS provides eventual consistency for new objects.
- **D.** S3 is suitable for immediate downloads because AWS provides strong consistency for new objects.

Explanation :

Answer: B

Amazon S3 provides read-after-write consistency for PUTS of new objects in your S3 bucket in all regions with one caveat. The caveat is that if you make a HEAD or GET request to the key name (to find if the object exists) before creating the object, Amazon S3 provides eventual consistency for read-after-write.

- **Read-after-write Consistency:** Amazon S3 now supports read-after-write consistency for new objects added to Amazon S3 in US Standard region. Prior to this announcement, all regions except US Standard supported read-after-write consistency for new objects uploaded to Amazon S3. With this enhancement, Amazon S3 now supports read-after-write consistency in all regions for new objects added to Amazon S3. Read-after-write consistency allows you to retrieve objects immediately after creation in Amazon S3.

Option A is not true. Eventual consistency is for overwrite PUTS and DELETES. Option C is not true. EFS provides read-after-write consistency.

Data Consistency in Amazon EFS

Amazon EFS provides the open-after-close consistency semantics that applications expect from NFS.

In Amazon EFS, write operations will be durably stored across Availability Zones when:

- An application performs a synchronous write operation (for example, using the `open` Linux command with the `O_DIRECT` flag, or the `fsync` Linux command).
- An application closes a file.

Amazon EFS provides stronger consistency guarantees than open-after-close semantics depending on the access pattern. Applications that perform synchronous data access and perform non-appending writes will have read-after-write consistency for data access.

For option D, AWS provides strong consistency for DynamoDB, not for S3.

DynamoDB supports *eventually consistent* and *strongly consistent* reads.

Eventually Consistent Reads

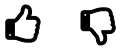
When you read data from a DynamoDB table, the response might not reflect the results of a recently completed write operation. The response might include some stale data. If you repeat your read request after a short time, the response should return the latest data.

Strongly Consistent Reads

When you request a strongly consistent read, DynamoDB returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful. A strongly consistent read might not be available if there is a network delay or outage.

> **Note**
>
> DynamoDB uses eventually consistent reads, unless you specify otherwise. Read operations (such as `GetItem`, `Query`, and `Scan`) provide a `ConsistentRead` parameter. If you set this parameter to true, DynamoDB uses strongly consistent reads during the operation.

Ask our Experts



QUESTION 5 CORRECT

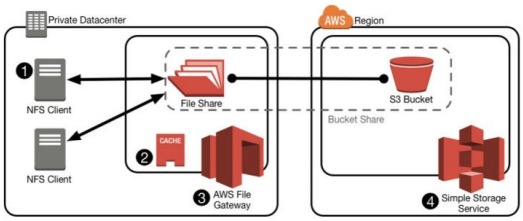
You are a solutions architect. Your organization is building an application on premise. But would like to keep the storage on AWS. Objects/files must only be accessed via the application as there are relational and access related logics built in the application. But, as an exception, Administrators should be able to access the objects/files directly from AWS S3 console/API bypassing the application. What solution would you provide?

- A. Cached Volume Gateway
- B. Stored Volume Gateway
- C. File Gateway ✔
- D. Custom built S3 solution

Explanation :

Answer: C

The File Gateway presents a file interface that enables you to store files as objects in Amazon S3 using the industry-standard NFS and SMB file protocols, and access those files via NFS and SMB from your datacenter or Amazon EC2, or access those files as objects with the S3 API.



- https://d1.awsstatic.com/whitepapers/aws-storage-gateway-file-gateway-for-hybrid-architectures.pdf (https://d1.awsstatic.com/whitepapers/aws-storage-gateway-file-gateway-for-hybrid-architectures.pdf)

For option A, with Cached Volumen Gateway, you store your data in Amazon Simple Storage Service (Amazon S3) and retain a copy of frequently accessed data subsets locally. However, these are stored as snapshots in S3 and cannot be accessed through console/API.

For option B, with stored volumes, you store the entire set of volume data on-premises and store periodic point-in-time backups (snapshots) in AWS. In this model, your on-premises storage is primary, delivering low-latency access to your entire dataset. AWS storage is the backup that you can restore in the event of a disaster in your data center.
For option C, although custom built solution using S3 might work, it is recommended to use
AWS provided services where ever possible.

- For more information in AWS storage gateways, refer documentation here. https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html (https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html)

Ask our Experts

QUESTION 6 CORRECT

You have created an S3 bucket in us-east-1 region with default configurations. You have uploaded few documents and would like to share it with a group of people in your organization within the specified time duration. What is the recommended approach?

○ **A.** Create one IAM user per person, attach managed policy for each user with GetObject action on your S3 bucket. Users can login to AWS console and download documents.

- **B.** Create one IAM user per person, add them to an IAM grop, attach managed policy for the group with GetObject action on your S3 bucket. Users can login to AWS console and download documents.
- **C.** Generate pre-signed URL with an expiry date and share the URL with all persons via email. ✔
- **D.** By default, S3 bucket has public access enabled. Share the document URLs with all persons via email.

Explanation :

Answer: C

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

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

For more information, refer documentation here.

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

For options A and B, although these solutions work, it's a whole lot of setup for enabling download of documents. Also, AWS recommends using temporary credentials for use cases where users ocasionally need access to AWS resources.
In this case, pre-signed URL is granting temporary access on the S3 objects and access gets expired when the time limit has reached.
Option D is incorrect. All objects in S3 bucket are private by default.

Ask our Experts

QUESTION 7 CORRECT

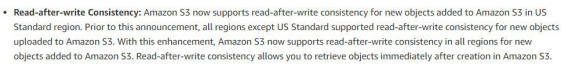
Which of the following are valid statements about Amazon S3? (Choose 3 options)

- [ ] **A.** S3 provides read-after-write consistency for any type of PUTS.
- [ ] **B.** S3 provides strong consistency for PUTs or DELETES.
- [x] **C.** A successful response to a PUT request for new object only occurs when the object is completely saved. ✔
- [x] **D.** S3 might return prior data when a process replaces an existing object and immediately attempts to read. ✔
- [x] **E.** S3 provides eventual consistency for overwrite PUTS and DELETES ✔

Explanation :

Answer: C, D, E

Amazon S3 provides read-after-write consistency for PUTS of new objects in your S3 bucket in all regions with one caveat. The caveat is that if you make a HEAD or GET request to the key name (to find if the object exists) before creating the object, Amazon S3 provides eventual consistency for read-after-write.



- **Read-after-write Consistency:** Amazon S3 now supports read-after-write consistency for new objects added to Amazon S3 in US Standard region. Prior to this announcement, all regions except US Standard supported read-after-write consistency for new objects uploaded to Amazon S3. With this enhancement, Amazon S3 now supports read-after-write consistency in all regions for new objects added to Amazon S3. Read-after-write consistency allows you to retrieve objects immediately after creation in Amazon S3.

Amazon S3 offers eventual consistency for overwrite PUTS and DELETES in all regions. For more information on S3 consistency model, refer documentation here.

https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html #CoreConcepts (https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#CoreConcepts)

and refer to "Amazon S3 Data Consistency Model"

Option A is incorrect. Read-after-write consistency is only provided for new object PUTS, not for any type of PUTS.

Option B is incorrect. AWS does not provide strong consistency for S3 objects. Strong consistency model is for DynamoDB reads.

Option C translates to read-after-write consistency model. Hence correct.

Option D translates to eventual consistency model. Hence correct. Option E is correct from above statements.

Ask our Experts

QUESTION 8 CORRECT

You are designing a web application that stores static assets in an Amazon S3 bucket. You expect this bucket to immediately receive over 400 requests with a mix of GET/PUT/DELETE per second. What should you do to ensure optimal performance?

- **A.** Amazon S3 will automatically manage performance at this scale. ✔
- **B.** Add a random prefix to the key names.
- **C.** Use a predictable naming scheme, such as sequential numbers or date time sequences, in the key names.
- **D.** Use multi-part upload.

Explanation :

Correct Answer: B

Latest Update: Based on the New S3 announcement (S3 performance)Amazon S3 now provides increased request rate performance. But AWS not yet updated the exam Questions. So as per exam Option B is the correct answer.

- https://aws.amazon.com/about-aws/whats-new/2018/07/amazon-s3-announces-increased-request-rate-performance/ (https://aws.amazon.com/about-aws/whats-new/2018/07/amazon-s3-announces-increased-request-rate-performance/)

Amazon S3 maintains an index of object key names in each AWS Region. Object keys are stored in UTF-8 binary ordering across multiple partitions in the index. The key name determines which

partition the key is stored in. Although Amazon S3 automatically scales to high request rates, using a sequential prefix, such as timestamp or an alphabetical sequence, increases the likelihood that Amazon S3 will target a specific partition for a large number of your keys, potentially overwhelming the I/O capacity of
the partition. When your workload is a mix of request types, introduce some randomness to key names by adding a hash string as a prefix to the key name. By introducing randomness to your key names the I/O load will be distributed across multiple index partitions. For example, you can compute an MD5 hash of the character sequence that you plan to assign as the key and add 3 or 4 characters from the hash as a prefix to the key name. The following example shows key names with a 4 character hexadecimal hash added as a prefix.

Without the 4 character hash prefix, S3 may distribute all of this load to 1 or 2 index partitions since the name of each object begins with examplebucket/2013-26-05-15-00-0 and all objects in
the index are stored in alpha-numeric order. The 4 character hash prefix ensures that the load is spread across multiple index partitions. When your workload is sending mostly GET requests, you can add randomness to key names. In addition, you can integrate Amazon CloudFront with Amazon S3 to distribute content to your users with low latency and a high data transfer rate.

```
examplebucket/232a-2013-26-05-15-00-00/cust1234234/photo1.jpg
examplebucket/7b54-2013-26-05-15-00-00/cust3857422/photo2.jpg
examplebucket/921c-2013-26-05-15-00-00/cust1248473/photo2.jpg
examplebucket/ba65-2013-26-05-15-00-00/cust8474937/photo2.jpg
examplebucket/8761-2013-26-05-15-00-00/cust1248473/photo3.jpg
examplebucket/2e4f-2013-26-05-15-00-01/cust1248473/photo4.jpg
examplebucket/9810-2013-26-05-15-00-01/cust1248473/photo5.jpg
examplebucket/7e34-2013-26-05-15-00-01/cust1248473/photo6.jpg
examplebucket/c34a-2013-26-05-15-00-01/cust1248473/photo7.jpg
...
```

Explanation based on the New announcement:

AWS Doc says that

Amazon S3 now provides increased performance to support at least 3,500 requests per second to add data and 5,500 requests per second to retrieve data, which can save significant processing time for no additional charge. Each S3 prefix can support these request rates, making it simple to increase performance significantly.

For More Information:

- https://aws.amazon.com/about-aws/whats-new/2018/07/amazon-s3-announces-increased-request-rate-performance/ (https://aws.amazon.com/about-aws/whats-new/2018/07/amazon-s3-announces-increased-request-rate-performance/)

Ask our Experts

QUESTION 9 CORRECT

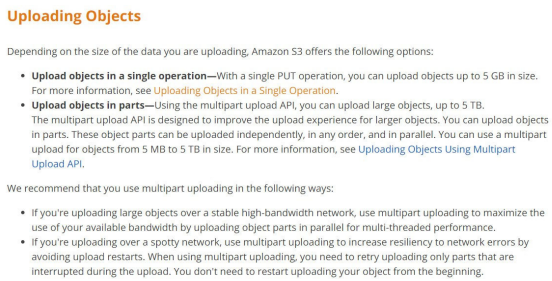
You have an applicaton running on EC2. When the application trying to upload a 7 GB file to S3, operation fails. What could be the reason for failure and what would be the solution?

- **A.** With a single PUT operation, you can upload objects up to 5 GB in size. Use multi-part upload for larger file uploads. ✔
- **B.** EC2 is designed to work best with EBS volumes. Use EBS Provisioned IOPs and use an Amazon EBS-optimized instance.
- **C.** NAT gateway only supports data transfers going out upto 5 GB. Use EBS Provisioned IOPs and use an Amazon EBS-optimized instance.
- **D.** VPC Endpoints only supports data transfers going out upto 5 GB. Use EBS Provisioned IOPs and use an Amazon EBS-optimized instance.

Explanation :

Answer: A

AWS recommends using multi-part uploads for larger objects.



Uploading Objects

Depending on the size of the data you are uploading, Amazon S3 offers the following options:

- **Upload objects in a single operation**—With a single PUT operation, you can upload objects up to 5 GB in size. For more information, see Uploading Objects in a Single Operation.
- **Upload objects in parts**—Using the multipart upload API, you can upload large objects, up to 5 TB. The multipart upload API is designed to improve the upload experience for larger objects. You can upload objects in parts. These object parts can be uploaded independently, in any order, and in parallel. You can use a multipart upload for objects from 5 MB to 5 TB in size. For more information, see Uploading Objects Using Multipart Upload API.

We recommend that you use multipart uploading in the following ways:

- If you're uploading large objects over a stable high-bandwidth network, use multipart uploading to maximize the use of your available bandwidth by uploading object parts in parallel for multi-threaded performance.
- If you're uploading over a spotty network, use multipart uploading to increase resiliency to network errors by avoiding upload restarts. When using multipart uploading, you need to retry uploading only parts that are interrupted during the upload. You don't need to restart uploading your object from the beginning.

For more information on multi-part uploads, refer documentation here.
https://docs.aws.amazon.com/AmazonS3/latest/dev/mpuoverview.html (https://docs.aws.amazon.com/AmazonS3/latest/dev/mpuoverview.html)

For option B, Amazon EBS is a storage for the drives of your virtual machines. It stores data as blocks of the same size and organizes them through the hierarchy similar to a traditional file system. EBS is not a standalone storage service like Amazon S3 so you can use it only in combination with Amazon EC2.
Objects can be stored on EBS volumes, but not cost-effective and not highly resilient and fault tolarant compared to S3.
Optionc C and D are incorrect. NAT Gateway ad VPC endpoints do not have any data transfer limitations.

Ask our Experts

QUESTION 10 **CORRECT**

You have an application on EC2 which stores the files in an S3 bucket. EC2 is being launched using a role which has GetObject permissions on the S3 bucket defined in its policy. The users who authenticate to this application

will get a pre-signed URL for the files in S3 bucket using EC2 role temporary credentials. However, users reporting they get an error when accessing pre- signed URLs. What could be the reason?(Choose 2 options)

- [x] **A.** Pre-signed URLs expired. ✔
- [ ] **B.** Logged in user must be an IAM user to download file through pre-signed URL.
- [x] **C.** Bucket might have a policy with Deny. EC2 role not whitelisted in the policy statement with Deny. ✔
- [ ] **D.** Default policy on temporary credentials does not have GetObject privileges on S3 bucket.

Explanation :

Answer: A, C

All objects in S3 are private by default. Only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a pre- signed URL, using their own security credentials, to grant time-limited permission to download the objects.

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

For more information, refer documentation here.
https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html (https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html)

For option A, while generating pre-signed URL programatically using SDK/API, we give a duration how long should the URL be valid. When the URL is accessed after the specified duration, you would get an error.

For option B, AWS recommends to use temporary credentials when ever users need time-limited access to AWS resources instead of using IAM users for each request.

For more information on temporary credentials, refer documentation here.
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html)

For option C, if a bucket policy contains Effect as Deny, you must whitelist all the IAM resources which need access on the bucket. Otherwise, IAM resources cannot access S3 bucket even if they have full access.

For detailed information on how to restrict bucket, refer documentation here. https://aws.amazon.com/blogs/security/how-to-restrict-amazon-s3-bucket-access-to-a-specific- (https://aws.amazon.com/blogs/security/how-to-restrict-amazon-s3-bucket-access-to-a-specific-iam-role/)iam-role/ (https://aws.amazon.com/blogs/security/how-to-restrict-amazon-s3-bucket-access-to-a-specific-)

For option D, policy is an optional parameter when temporary credentials are generated using AssumeRole (which is how EC2 generates temporary credentials using instance-profile). There is no default policy.