









[Differences between Amazon SQS, Amazon MQ, and Amazon SNS - Amazon Simple Queue Service](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-difference-from-amazon-mq-sns.html)

[CreateQueue - Amazon Simple Queue Service](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_CreateQueue.html)

[Amazon API Gateway concepts - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-basic-concept.html)

[Deploying a REST API in Amazon API Gateway - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-deploy-api.html)

[Set up an API Gateway canary release deployment - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/canary-release.html)

[Compute – Amazon EC2 Instance Types – AWS](https://aws.amazon.com/ec2/instance-types/)

<https://aws.amazon.com/blogs/aws/new-application-load-balancer-sni/>

[Using server-side encryption with AWS KMS keys (SSE-KMS) - Amazon Simple Storage Service](https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingKMSEncryption.html)

[What is Amazon Simple Queue Service? - Amazon Simple Queue Service](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html)

<https://aws.amazon.com/secrets-manager/>

<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-paramstore.html>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-basic-concept.html>

[Deploying a REST API in Amazon API Gateway - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-deploy-api.html)

[Use API Gateway Lambda authorizers - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-use-lambda-authorizer.html)

[Advanced environment customization with configuration files (.ebextensions) - AWS Elastic Beanstalk (amazon.com)](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ebextensions.html)

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/AWSHowTo.RDS.html>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ebextensions.html>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-environment-resources-elasticache.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>

[Caching Best Practices | Amazon Web Services](https://aws.amazon.com/caching/best-practices/)

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features-managing-env-tiers.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-short-and-long-polling.html>

<https://aws.amazon.com/blogs/database/work-with-cluster-mode-on-amazon-elasticache-for-redis/>

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

<https://aws.amazon.com/kinesis/data-streams/>

<https://aws.amazon.com/sqs/>

<https://aws.amazon.com/kinesis/data-firehose/>

Question 1: **Incorrect**

**Your organization wants to store customer information in Amazon S3 with encryption at rest. You want to manage the encryption CMKs in an AWS Managed Service. Which of these options would you recommend?**



**Use Server Side Encryption with Amazon S3 managed keys**

**(Incorrect)**



**Use Server Side Encryption with AWS KMS**

**(Correct)**



**Use Server Side Encryption with Customer provided Keys**



**Use Client Side Encryption with Customer provided Keys**

Explanation

SSE-S3:

AWS S3 manages its own keys

Keys are rotated every month

Request Header - x-amz-server-side-encryption(AES256)

SSE-KMS:

Customer manages keys in KMS

Request Headers - x-amz-server-side-encryption(aws:kms) and x-amz-server-side-encryption-aws-kms-key-id(ARN for key in KMS)

SSE-C:

Customer sends the key with every request

S3 performs encryption and decryption without storing the key

HTTPS is mandatory

Question 15: **Incorrect**

**You are working for a major bank using an Amazon S3 Bucket to store your files. You want to encrypt files with your own Customer Master Keys. However, you do not want to worry about the encryption process. Which of these S3 Encryption approaches would you recommend?**



**Client-Side Encryption with Customer-Provided Keys**



**Server-Side Encryption with Customer-Provided Keys (SSE-C)**

**(Correct)**



**Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)**

**(Incorrect)**



**Server-Side Encryption with Amazon Amazon S3-Managed Keys (SSE-Amazon S3)**

Explanation

**Server-Side Encryption - S3**

**SSE-S3**:

AWS S3 manages its own keys

Keys are rotated every month

Request Header - x-amz-server-side-encryption(AES256)

**SSE-KMS**:

The customer manages keys in KMS

Request Headers - x-amz-server-side-encryption(aws:kms) and x-amz-server-side-encryption-aws-kms-key-id(ARN for key in KMS)

**SSE-C**:

The customer sends the key with every request

S3 performs encryption and decryption without storing the key

HTTPS is mandatory

Question 26: **Correct**

A security company is requiring all developers to perform server-side encryption with customer-provided encryption keys when performing operations in AWS S3. Developers should write software with C# using the AWS SDK and implement the requirement in the PUT, GET, Head, and Copy operations.

Which of the following encryption methods meets this requirement?



**Client-Side Encryption**



**SSE-KMS**



**SSE-C**

**(Correct)**



**SSE-S3**

Explanation

Correct option:

**SSE-C**

You have the following options for protecting data at rest in Amazon S3:

Server-Side Encryption – Request Amazon S3 to encrypt your object before saving it on disks in its data centers and then decrypt it when you download the objects.

Client-Side Encryption – Encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

For the given use-case, the company wants to manage the encryption keys via its custom application and let S3 manage the encryption, therefore you must use Server-Side Encryption with Customer-Provided Keys (SSE-C).

Using server-side encryption with customer-provided encryption keys (SSE-C) allows you to set your encryption keys. With the encryption key you provide as part of your request, Amazon S3 manages both the encryption, as it writes to disks, and decryption, when you access your objects.

Please review these three options for Server Side Encryption on S3: A close-up of a computer error

Description automatically generated via - <https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

Incorrect options:

**SSE-KMS** - Server-Side Encryption with Customer Master Keys (CMKs) stored in AWS Key Management Service (SSE-KMS) is similar to SSE-S3. SSE-KMS provides you with an audit trail that shows when your CMK was used and by whom. Additionally, you can create and manage customer-managed CMKs or use AWS managed CMKs that are unique to you, your service, and your Region.

**Client-Side Encryption** - You can encrypt the data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

**SSE-S3** - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. As an additional safeguard, it encrypts the key itself with a master key that it regularly rotates. So this option is incorrect.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

[Customizing at the edge with functions - Amazon CloudFront](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/edge-functions.html)