PEARSON IT CERTIFICATION

Practice Tests

Flash Cards

Review Exercises

Study Planner

Cert Guide

Advance your IT career with hands-on learning

AWS Certified Solutions Architect – Associate

(SAA-C03)

MARK WILKINS

AWS Certified Solutions Architect – Associate (SAA-C03) Cert Guide

Access interactive study tools on this book's companion website, including practice test software, review questions, Key Term flash card application, a study planner, and more!

To access the companion website, simply follow these steps:

1. Go to **www.pearsonITcertification.com/register**.
2. Enter the print book ISBN: **9780137941582**.
3. Answer the security question to validate your purchase.
4. Go to your account page.
5. Click on the **Registered Products** tab.
6. Under the book listing, click on the Access Bonus Content link.

If you have any issues accessing the companion website, you can contact our support team by going to **http://pearsonitp.echelp.org.**

Pearson Test Prep online system requirements:
Browsers: Chrome version 73 and above; Safari version 12 and above; Microsoft Edge 44 and above.
Devices: Desktop and laptop computers, tablets running Android v8.0 and above or iPadOS v13 and above, smartphones running Android v8.0 and above or iOS v13 and above with a minimum screen size of 4.7". Internet access required.

Pearson Test Prep offline system requirements:
Windows 10, Windows 8.1; Microsoft .NET Framework 4.5 Client; Pentium-class 1 GHz processor (or equivalent); 512 MB RAM; 650 MB disk space plus 50 MB for each downloaded practice exam; access to the Internet to register and download exam databases.

Amazon S3 Bucket Security

By default, only the owner who created an S3 bucket has access to the objects stored in the bucket. There are several methods for controlling security for an S3 bucket (see Figure 5-9):

- **ACLs:** You can use ***access control lists (ACLs)*** to control primary access from other AWS accounts for list and write objects and read and write bucket permissions, public access, and access to S3 logging information. ACLs are available for purposes of backward compatibility and are the weakest type of S3 security (and therefore not recommended).

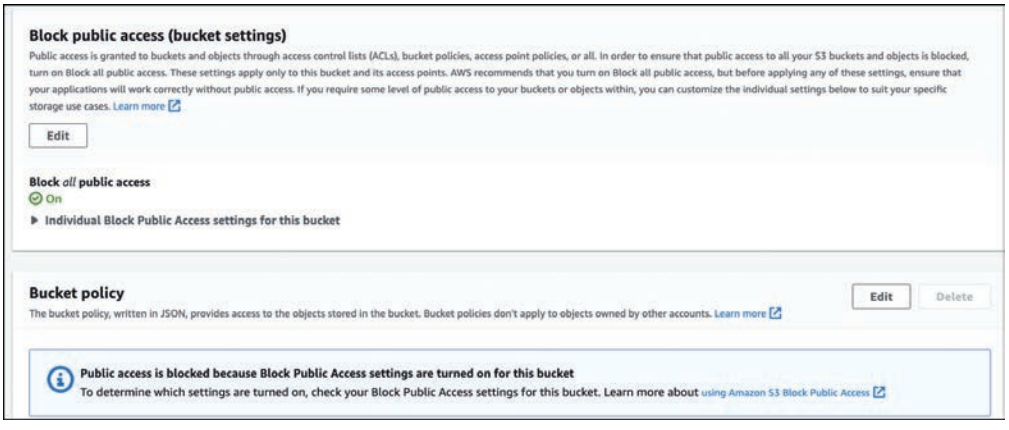

Figure 5-9 S3 Permission Settings

- **IAM policy:** You can grant access to other AWS users and groups of IAM users by using IAM permission policies in partnership with resource policies.
- **S3 Bucket policy:** You can control direct access to an S3 bucket, as shown in Example 5-2, by creating a ***bucket policy*** assigned directly to the S3 bucket. An S3 bucket policy is a JSON-formatted document that defines which actions are allowed or denied on an S3 bucket and its contents. A bucket policy is attached directly to the bucket it is protecting, and the policy settings list who has access to the bucket and what they can do with the objects in the bucket. An S3 bucket policy might allow a specific IAM user to read and write objects in the bucket, while denying access to all other users. Or, the policy might allow any user to read objects in the bucket but allow only authenticated users to write objects.

S3 bucket policies are defined using the AWS Policy Language, which provides a set of keywords and operations that you can use to specify the conditions under which a policy takes effect. A bucket policy can also allow access from multiple AWS accounts to a single S3 bucket.

Example 5-2 S3 Bucket Policy

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
                     "arn:aws:s3:::2021232reports",
         "arn:aws:s3:::2021232reports/*"
      ],
      "Condition": {
          "NotIpAddress": {"aws:SourceIp": "54.242.144.0/24"}
      }
    }
  ]
}
```

- **Query string authentication:** Query string authentication is a method to authenticate requests to an Amazon S3 bucket allowing organizations to generate a URL (see Figure 5-10) that can be shared with end users. When an end user clicks the URL, they are granted access to the specified S3 bucket and its contents.

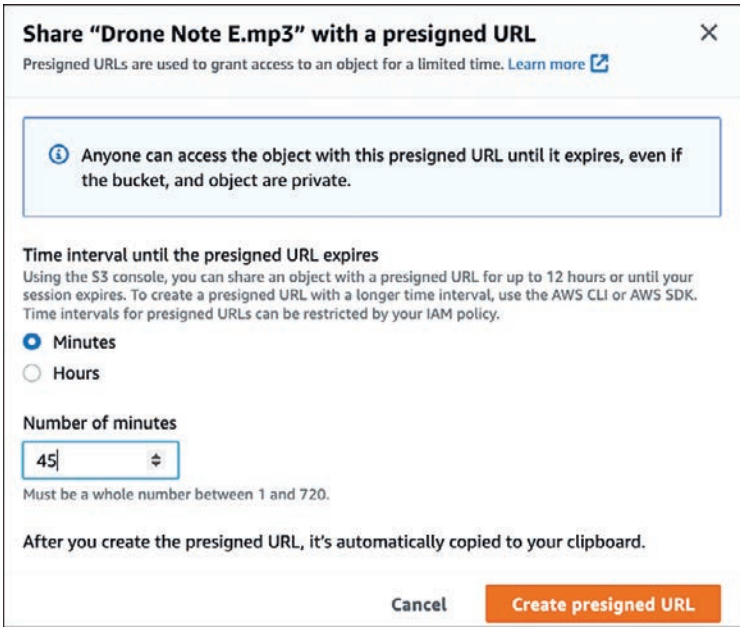

Figure 5-10 Presigned URL for S3 Object Access

The URL includes a set of parameters that specify the credentials that grant access to the bucket. These parameters include the access key ID, an expiration time for the URL, and a signature that is calculated using the access key secret.

When someone attempts to access the URL, the Amazon S3 service checks the signature to verify that it matches the expected value. If the signature is valid, the user is granted access to the bucket; otherwise, the request is denied.

The use case for using query string authentication is useful for granting temporary access to an S3 bucket without having to create an IAM user or provide AWS access keys. However, query string authentication is not as secure as IAM policies or bucket policies because the URL and its parameters are included in each request; therefore, anyone who has access to the URL can potentially gain access to the bucket.

NOTE If you require public access to objects in an S3 bucket, it's recommended that you create a separate AWS account specifically for hosting the S3 buckets that will have public S3 object access.

- **Blocking S3 public access:** S3 Buckets always start as private, with no default public access (see Figure 5-11). When the Block Public Access (Bucket Settings) setting is enabled, attempts at changing security settings to allow public access to objects in the S3 bucket are denied. You can block public access on an individual S3 bucket or on all S3 buckets in your AWS account by editing the public access settings for your account using the S3 console. Choices for blocking S3 public access include the following:
  - **Public:** Everyone has access to list objects, write objects, and read and write permissions.
  - **Objects Can Be Public:** The bucket is not public; however, public access can be granted to individual objects by users with permissions.
  - **Buckets and Objects Not Public:** No public access is allowed to the bucket or the objects within the bucket.

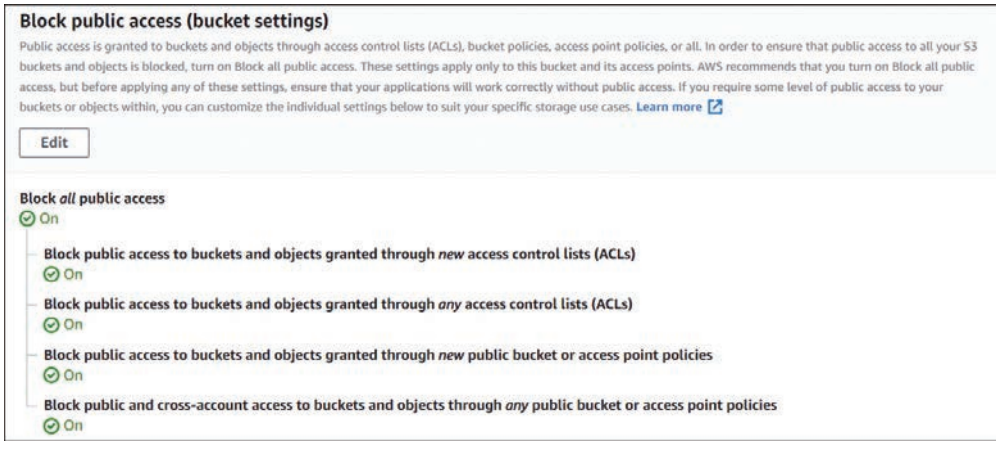

Figure 5-11 Blocking Public Access on an S3 Bucket by Default

NOTE Amazon Macie is a powerful AWS security service that uses artificial intelligence (AI) and machine learning (ML) technology to analyze your S3 objects and access patterns. Amazon S3 data can be classified based on many file formats, such as Personally Identifiable Information (PII) and other file types. AWS SNS notifications can be generated by Amazon Macie when Amazon S3 objects are discovered to be compromised.

Key Topic

S3 Storage at Rest

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, the key topics to know about S3 storage at rest are as follows:

- **SSE-S3:** With SSE-S3, Amazon S3 manages the encryption and decryption of the data in the bucket. Organizations that select this option don't manage the encryption keys but can access the data in the bucket without having to manage the keys. SSE-S3 uses the Advanced Encryption Standard (AES) algorithm with a 256-bit key to encrypt the data in the bucket. The key is automatically generated by Amazon S3 and is regularly rotated to ensure the security of the encrypted data (see Figure 5-12). Note that SSE encrypts the object data but the optional tag object metadata remains unencrypted.

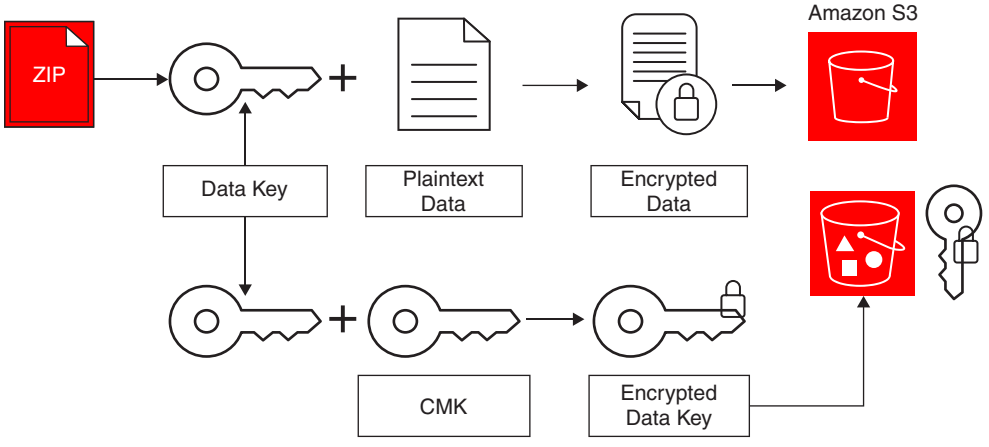

Figure 5-12 SSE-S3 Encryption Process

- **SSE-KMS:** Organizations can select AWS KMS to manage their encryption keys. Select the default CMK or choose a CMK that was already created in AWS KMS before starting an S3 encryption process. Accessing encrypted objects managed by KMS can be expensive: If you have an exceptionally large number of encrypted objects, a large volume of decryption requests will be made to KMS. You can configure SSE-KMS to significantly reduce the cost of the encryption and decryption process. When an S3 Bucket Key is configured for SSE-KMS server-side encryption, a short-lived encryption key is created and stored and used to encrypt objects internally inside AWS S3 rather than utilize AWS KMS encryption processes. The S3 Bucket Key creates unique data keys for encrypting objects in the specific S3 bucket that has enabled the S3 Bucket Key option. The encryption process reduces AWS KMS requests

for external encryption keys and can reduce encryption costs by 99%. The S3 Bucket Key is a worker process within the S3 bucket that enables you to perform encryption services without constant communication with KMS.

- **SSE-C:** You can use SSE with a customer-provided encryption key. With each request, the encryption key is provided to AWS, and Amazon S3 manages the encryption and decryption of S3 objects by using the supplied key. The same encryption key that was used to encrypt the object must be provided before the object can be decrypted (see Figure 5-13). After the encryption process is complete, the supplied encryption key is deleted from memory. To upload an object with an organization-provided encryption key (SSE-C), the AWS CLI, AWS SDK, or Amazon S3 REST API must be used.

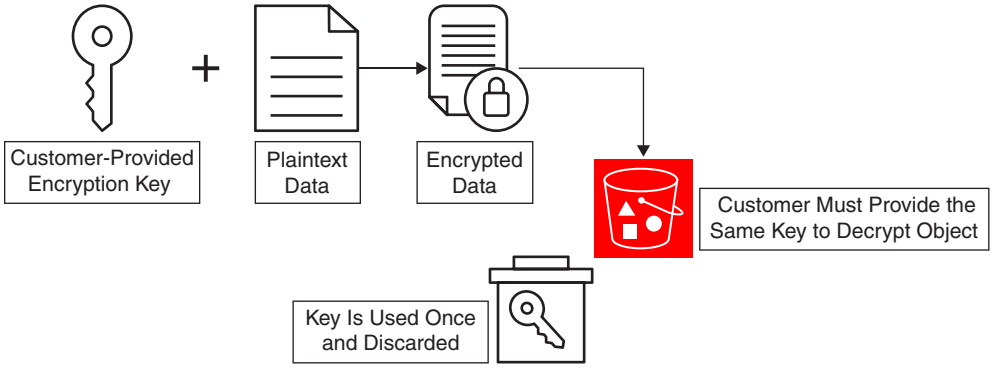

Figure 5-13 SSE-C Encryption Process

Key Topic

Amazon S3 Object Lock Policies

Amazon S3 buckets and Amazon S3 Glacier have data policies that can lock objects so they cannot be deleted or changed. Amazon S3 objects can be locked using a ***write-once/read-many (WORM)*** policy. Object lock policies enable you to set rules that restrict certain actions on objects, such as deleting or overwriting them, in order to protect objects and ensure they remain available and unaltered. Object lock policies are set at the S3 bucket level and apply to all objects in the bucket, or set on individual objects. This can be useful for complying with legal or regulatory requirements or protecting important or sensitive data. Apply a WORM policy, as shown in Figure 5-14, to stop an Amazon S3 object from being overwritten, or deleted for a fixed time period, or indefinitely. There are several options to WORM policies to understand. First is the *retention period*, which refers to a set number of days or years during which an object will remain