review

review questions

Simple Storage Service (S3)

V1.02

Course title

BackSpace Academy
AWS Certified Associate

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: Managing Access Permissions to Your Amazon S3 Resources

Reference: Amazon Simple Storage Service Developer Guide

https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html

Question

Object policies and access control lists (ACLs) are resource-based because you attach them to your Amazon S3 resources.

Answers

A. True
B. False

B
Bucket policies not object policies.
Amazon S3 offers access policy options broadly categorized as resource-based policies and user policies. Access policies you attach to your resources (buckets and objects) are referred to as resource-based policies. For example, bucket policies and access control lists (ACLs) are resource-based policies. You can also attach access policies to users in your account. These are called user policies. You may choose to use resource-based policies, user policies, or some combination of these to manage permissions to your Amazon S3 resources. The introductory topics provide general guidelines for managing permissions.

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html

Question

Using IAM, you can create IAM users, groups, and roles in your account and attach access policies to them granting them access to AWS resources including Amazon S3.

Answers

A. True
B. False

A
Bucket policy and user policy are two of the access policy options available for you to grant permission to your Amazon S3 resources. Both use JSON-based access policy language.

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/example-policies-s3.html

Question

To perform a specific operation on an object, an IAM user needs permission from the parent AWS account to which it belongs or the AWS account that owns the object.

Answers

A. True
B. False

B
When Amazon S3 receives a request for an object operation, it converts all the relevant permissions—resource-based permissions (object access control list (ACL), bucket policy, bucket ACL) and IAM user policies—into a set of policies to be evaluated at run time. It then evaluates the resulting set of policies in a series of steps. In each step, it evaluates a subset of policies in three specific contexts—user context, bucket context, and object context.
User context – If the requester is an IAM user, the user must have permission from the parent AWS account to which it belongs.

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/access-control-auth-workflow-object-operation.html

Question

An object ACL is preferred:

Answers

A. When you want to manage access to objects not owned by the bucket owner or when permissions vary by object and you need to manage permissions at the object level
B. When you want to manage cross-account permissions for all Amazon S3 permissions
C. When permission from the parent account to a user is required
D. When you want to grant write permission to the Amazon S3 Log Delivery group to write access log objects to your bucket

A
When to Use an Object ACL
In addition to an object ACL, there are other ways an object owner can manage object permissions. For example:

If the AWS account that owns the object also owns the bucket, then it can write a bucket policy to manage the object permissions.

If the AWS account that owns the object wants to grant permission to a user in its account, it can use a user policy.

So when do you use object ACLs to manage object permissions? The following are the scenarios when you use object ACLs to manage object permissions:

- An object ACL is the only way to manage access to objects not owned by the bucket owner
- Permissions vary by object and you need to manage permissions at the object level
- Object ACLs control only object-level permissions

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/access-policy-alternatives-guidelines.html#when-to-use-acl

Question

A Bucket Policy is preferred:

Answers

A. When you want to manage access to objects not owned by the bucket owner
B. When you want to manage cross-account permissions for all Amazon S3 permissions
C. When permission from the parent account to a user is required
D. When you want to grant write permission to the Amazon S3 Log Delivery group to write access log objects to your bucket
E. When permissions vary by object and you need to manage permissions at the object level

B
If an AWS account that owns a bucket wants to grant permission to users in its account, it can use either a bucket policy or a user policy. But in the following scenario, you will need to use a bucket policy.

You want to manage cross-account permissions for all Amazon S3 permissions

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/access-policy-alternatives-guidelines.html#when-to-use-bucket-policy

Question

A User Policy is preferred:

Answers

A. When you want to manage access to objects not owned by the bucket owner
B. When you want to manage cross-account permissions for all Amazon S3 permissions
C. When you want to grant write permission to the Amazon S3 Log Delivery group to write access log objects to your bucket
D. When permission from the parent account to a user is required
E. When permissions vary by object and you need to manage permissions at the object level

D
WS Identity and Access Management (IAM) enables you to create multiple users within your AWS account and manage their permissions via user policies. An IAM user must have permissions from the parent account to which it belongs, and from the AWS account that owns the resource the user wants to access. The permissions can be granted as follows:

- Permission from the parent account – The parent account can grant permissions to its user by attaching a user policy.
- Permission from the resource owner – The resource owner can grant permission to either the IAM user (using a bucket policy) or the parent account (using a bucket policy, bucket ACL, or object ACL).

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/access-policy-alternatives-guidelines.html#when-to-use-user-policy

Question

A Bucket ACL is preferred:

Answers

A. When you want to manage access to objects not owned by the bucket owner
B. When you want to manage cross-account permissions for all Amazon S3 permissions
C. When you want to grant write permission to the Amazon S3 Log Delivery group to write access log objects to your bucket
D. When permission from the parent account to a user is required
E. When permissions vary by object and you need to manage permissions at the object level

C
The only recommended use case for the bucket ACL is to grant write permission to the Amazon S3 Log Delivery group to write access log objects to your bucket (see Server Access Logging). If you want Amazon S3 to deliver access logs to your bucket, you will need to grant write permission on the bucket to the Log Delivery group. The only way you can grant necessary permissions to the Log Delivery group is via a bucket ACL.

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/access-policy-alternatives-guidelines.html#when-to-use-acl

Question

In its most basic sense, a policy contains the following elements:

- Resources
- Actions
- Effect
- Principal
- Object

Answers

A. True
B. False

B
Resources Actions Effect Principal

Question

What is the resource arn:aws:s3:::mybucket/developers/${aws:username}/ ?

Answers

A. An object in mybucket/developers containing aws or username in its prefix.
B. A folder in mybucket/developers with name the same as the user name.
C. A folder called mybucket/developers belonging to aws account username.

B
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/example-policies-s3.html#iam-policy-ex0

Question

What does the following bucket policy do?

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "statement1",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam:: 123456789012:user/Bill"
   },
   "Action": "s3:*",
   "Resource": "arn:aws:s3:::examplebucket/*"
  }
 ]
}
```

Answers

A. Grants the s3:PutObject and the s3:PutObjectAcl permissions to a user (Bill)
B. Grants all S3 action permissions to a user (Bill)
C. Reports a syntax error

B
Action s3:*

Question

The access policy language allows you to specify _____ when granting permissions. The _____element (or _____block) lets you specify _____for when a policy is in effect. In the _____element, which is optional, you build expressions in which you use Boolean operators (equal, less than, etc.) to match your _____against values in the request.

Answers

A. Objects
B. Policies
C. Conditions
D. Resources

C
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/amazon-s3-policy-keys.html

Question

What does the following bucket policy do?

```
{
 "Version": "2012-10-17",
 "Id": "S3PolicyId1",
 "Statement": [
  {
   "Sid": "IPAllow",
   "Effect": "Allow",
   "Principal": "*",
   "Action": "s3:*",
   "Resource": "arn:aws:s3:::examplebucket/*",
   "Condition": {
    "IpAddress": {
     "aws:SourceIp": "54.240.143.0/24"
    },
    "NotIpAddress": {
     "aws:SourceIp": "54.240.143.188/32"
    }
   }
  }
 ]
}
```

Answers

A. Grants permissions to any user to perform any Amazon S3 operations on objects in the examplebucket bucket. However, the request must originate from the range of IP addresses specified in 54.240.143.0/24 but excluding 54.240.143.188/32.
B. Grants permissions to any user to perform any Amazon S3 operations on objects in the examplebucket bucket.
C. Grants permissions to any user to perform any Amazon S3 operations on objects in the examplebucket bucket. However, the request must not originate from the range of IP addresses specified in 54.240.143.0/24 but excluding 54.240.143.188/32.

A
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-3

Question

The following policy denies any Amazon S3 operation on the /taxdocuments folder in the examplebucket bucket if the request is not MFA authenticated.?

```
{
 "Version": "2012-10-17",
 "Id": "123",
 "Statement": [
  {
   "Sid": "",
   "Effect": "Deny",
   "Principal": "*",
   "Action": "s3:*",
   "Resource": "arn:aws:s3:::examplebucket/taxdocuments/*",
   "Condition": {
    "Null": {
     "aws:MultiFactorAuthAge": true
    }
   }
  }
 ]
}
```

Answers

A. True
B. False

A
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-7

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: Protecting Data in Amazon S3

Reference: Amazon Simple Storage Service Developer Guide

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

Question

You have the following options of protecting data at rest in Amazon S3.

Answers

A. Use Server-Side Encryption
B. Use Client-Side Encryption
C. All of the above

C
Data protection refers to protecting data while in-transit (as it travels to and from Amazon S3) and at rest (while it is stored on disks in Amazon S3 data centers). You can protect data in transit by using SSL or by using client-side encryption. You have the following options of protecting data at rest in Amazon S3.

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

Use Client-Side Encryption – You can 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.

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

Question

You use Server-Side Encryption with:

Answers

A. Amazon S3-Managed Keys (SSE-S3)
B. AWS KMS-Managed Keys (SSE-KMS)
C. Customer-Provided Keys (SSE-C)
D. All of the above

D
Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a presigned URL, that URL works the same way for both encrypted and unencrypted objects.

Note
You can't apply different types of server-side encryption to the same object simultaneously.
You have three mutually exclusive options depending on how you choose to manage the encryption keys.

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

Question

Amazon S3 server-side encryption uses one of the strongest block ciphers available, 128-bit Advanced Encryption Standard (AES-128), to encrypt your data.

Answers

A. True
B. False

B
Not 128 bit
Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data.
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html

Question

Presigned URLs do not support the SSE-C.

Answers

A. True
B. False

B
You can generate a presigned URL that can be used for operations such as upload a new object, retrieve an existing object, or object metadata. Presigned URLs support the SSE-C as follows:

- When creating a presigned URL, you must specify the algorithm using the x-amz-server-side-encryption-customer-algorithm in the signature calculation.
- When using the presigned URL to upload a new object, retrieve an existing object, or retrieve only object metadata, you must provide all the encryption headers in your client application.

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerSideEncryptionCustomerKeys.html#ssec-and-presignedurl

Question

Once you version-enable a bucket, it can never return to an unversioned state.

Answers

A. True
B. False

A
Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.

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

Question

Objects stored in your bucket before you set the versioning state have a version ID of_____.

Answers

A. 0
B. Null
C. Unversioned

B
Objects stored in your bucket before you set the versioning state have a version ID of null. When you enable versioning, existing objects in your bucket do not change. What changes is how Amazon S3 handles the objects in future requests
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html

Question

You can optionally add another layer of security by configuring a bucket to enable _____Delete, which requires additional authentication for either changing the versioning state of your bucket or permanently deleting an object version.

Answers

A. IAM
B. Disable
C. MFA

C
You can optionally add another layer of security by configuring a bucket to enable MFA (Multi-Factor Authentication) Delete, which requires additional authentication for either of the following operations.

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

MFA Delete requires two forms of authentication together:

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

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

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

Question

To permanently delete versioned S3 objects, you must use DELETE Object versionId.

Answers

A. True
B. False

A
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html

Question

Once you suspend versioning on a bucket, Amazon S3 automatically adds a null version ID to every subsequent object stored thereafter in that bucket.

Answers

A. True
B. False

A
See: https://docs.aws.amazon.com/AmazonS3/latest/dev/AddingObjectstoVersionSuspendedBuckets.html

Question

If you delete the S3 delete marker, the object reappears in the console object list.

Answers

A. True
B. False

A
A delete marker is a placeholder (marker) for a versioned object that was named in a simple DELETE request. Because the object was in a versioning-enabled bucket, the object was not deleted. The delete marker, however, makes Amazon S3 behave as if it had been deleted.

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

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: Managing Access Permissions to Your Amazon S3 Resources

Reference: Amazon Simple Storage Service Developer Guide

https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html

Question

Amazon S3 supports the following destinations where it can publish events:

- Amazon SNS topic
- Amazon SQS queue
- AWS Lambda
- AWS SES email

Answers

A. True
B. False

B
Amazon S3 supports the following destinations where it can publish events:

- Amazon Simple Notification Service (Amazon SNS) topic
- Amazon Simple Queue Service (Amazon SQS) queue
- AWS Lambda function

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

Question

Amazon S3 can publish events of the following types.

- s3:ObjectCreated:
- s3:ObjectCreated:Put
- s3:ObjectCreated:Copy
- s3:ObjectDeleted
- s3:ReducedRedundancyLostObject

Answers

A. True
B. False

B

- s3:ObjectCreated:
- s3:ObjectCreated:Put
- s3:ObjectCreated:Copy
- s3:ReducedRedundancyLostObject

See: https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html#notification-how-to-event-types-and-destinations

This "learning by quizzes" exercise will be based upon the course videos and the following reference material:

Section: Performance Optimization

Reference: Amazon Simple Storage Service Developer Guide

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

Question

If your workload is mainly sending GET requests, you should consider using Amazon Elasticache for performance optimization.

Answers

A. True
B. False

B

If you want higher transfer rates over a single HTTP connection or single-digit millisecond latencies, use Amazon CloudFront or Amazon ElastiCache for caching with Amazon S3.