AWS S3 Permissions

# 

# S3 Permissions

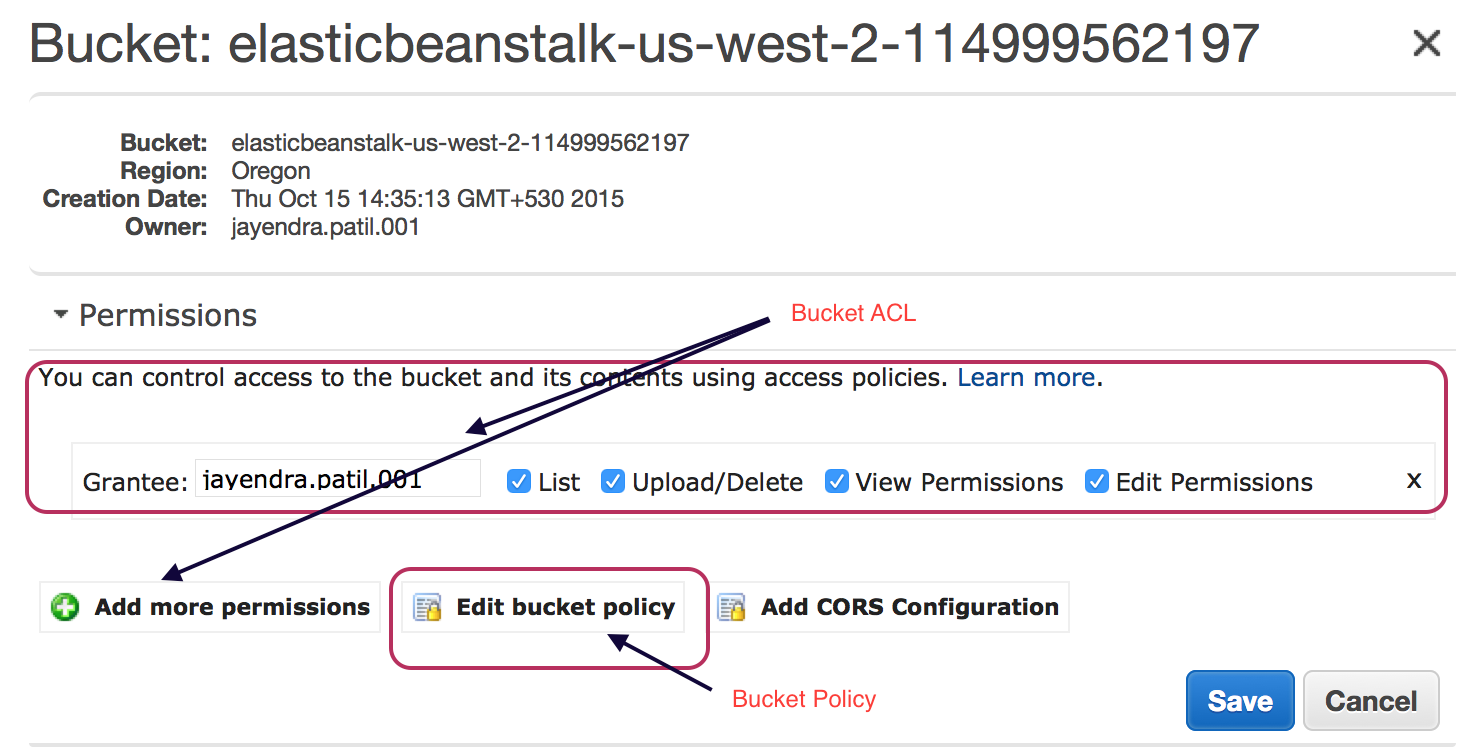
* By default, all S3 buckets, objects and related subresources are private
* User is the AWS Account or the IAM user who access the resource
* Bucket owner is the AWS account that created a bucket
* Object owner is the AWS account that uploads the object to a bucket, that means in a bucket other accounts can upload objects & they would be owner for that object.
* Only the Resource owner, the AWS account that creates the resource, can access the resource
* Resource owner can be
  + AWS account that creates the bucket or object owns those resources
  + If an IAM user creates the bucket or object, the AWS account of the IAM user owns the resource
  + If the bucket owner grants cross-account permissions to other AWS account users to upload objects to the buckets, the objects are owned by the AWS account of the user who uploaded the object and not the bucket owner except for the following conditions
    - Bucket owner can deny access to the object, as its still the bucket owner who pays for the object
    - Bucket owner can delete or apply archival rules to the object and perform restoration
* S3 permissions are classified into Resource based policies and User policies

## User policies

* User based policies use IAM with S3 to control the type of access a user or group of users has to specific parts of an S3 bucket the AWS account owns
* User based policy is always attached to an User, Group or a Role, anonymous permissions cannot be granted
* 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

## Resource based policies

Bucket policies and access control lists (ACLs) are resource-based because they are attached to the  S3 resources

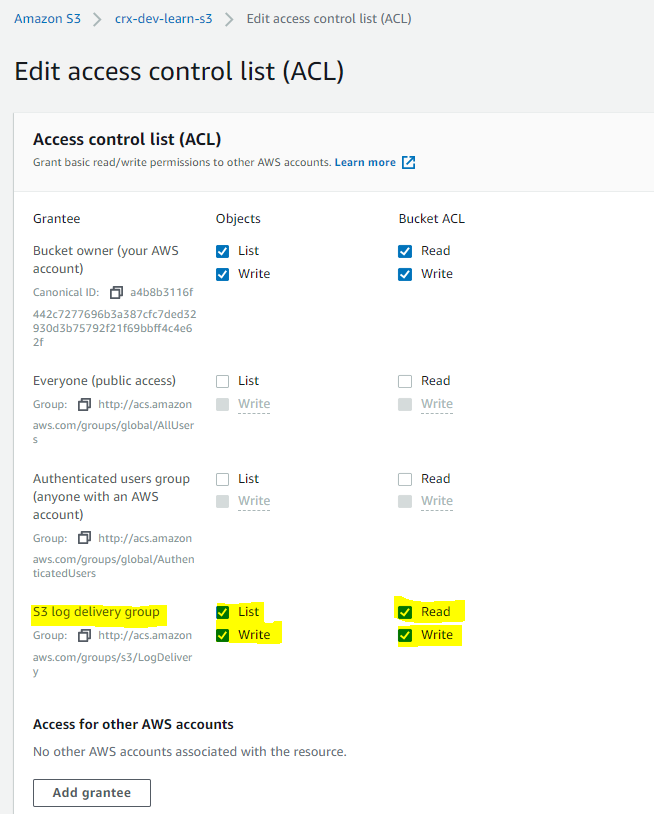


### Bucket Policies

* Bucket policy can be used to grant cross-account access to **other AWS accounts or IAM users in other accounts** for the bucket and objects in it.
* Bucket policies provide centralized, access control to buckets and objects based on a variety of conditions, including S3 operations, requesters, resources, and aspects of the request (e.g. IP address)
* 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
* Permissions attached to a bucket apply to all of the objects in that bucket created and owned by the bucket owner. Objects owned by other accounts would be excluded & needs to manage with Object ACL.
* Policies can either add or deny permissions across all (or a subset) of objects within a bucket
* Only the bucket owner is allowed to associate a policy with a bucket

### Access Control Lists (ACLs)

* **Each bucket and object have an ACL associated with it**.
* An ACL is a list of grants identifying grantee and permission granted
* **ACLs are used to grant basic read/write permissions on resources to other AWS accounts.**
* **ACL supports limited permissions set and**
  + **cannot grant conditional permissions, nor can you explicitly deny permissions**
  + cannot be used to grant permissions for bucket subresources
* Permission can be granted to an AWS account by the email address or the canonical user ID (is just an obfuscated Account Id). If an email address is provided, S3 will still find the canonical user ID for the user and add it to the ACL.
* It is Recommended to use Canonical user ID as email address would not be supported
* **Bucket ACL**
  + Only recommended use case for the bucket ACL is to grant write permission to S3 Log Delivery group to write access log objects to the bucket
  + Bucket ACL will help grant write permission on the bucket to the Log Delivery group if access log delivery is needed to your bucket
  + **Only way you can grant necessary permissions to the Log Delivery group is via a bucket ACL**



* **Object ACL**
  + **Object ACLs control only Object-level Permissions**
  + Object ACL is the **only way to manage permission to an object in the bucket not owned by the bucket owner** i.e. If the bucket owner allows cross-account object uploads and if the object owner is different from the bucket owner, the only way for the object owner to grant permissions on the object is through Object ACL
  + If the Bucket and Object is owned by the same AWS account, Bucket policy can be used to manage the permissions
  + If the Object and User is owned by the same AWS account, User policy can be used to manage the permissions

### Amazon S3 Request Authorization

When S3 receives a request, it must evaluate all the user policies, bucket policies and acls to determine whether to authorize or deny the request.

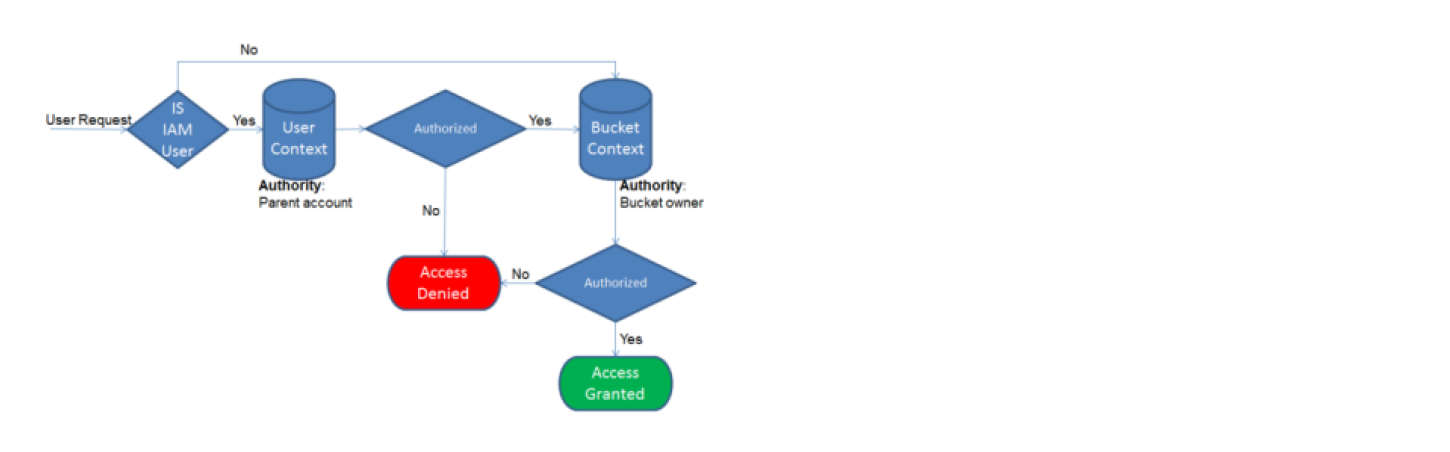
#### **S3 evaluates the policies in 3 context**

* **User context** is basically the context in which S3 evaluates the User policy that the parent AWS account (context authority) attaches to the user
* **Bucket context** is the context in which S3 evaluates the access policies owned by the bucket owner (context authority) to check if the bucket owner has not explicitly denied access to the resource
* **Object context** is the context where S3 evaluates policies owned by the Object owner (context authority)

#### **Analogy**

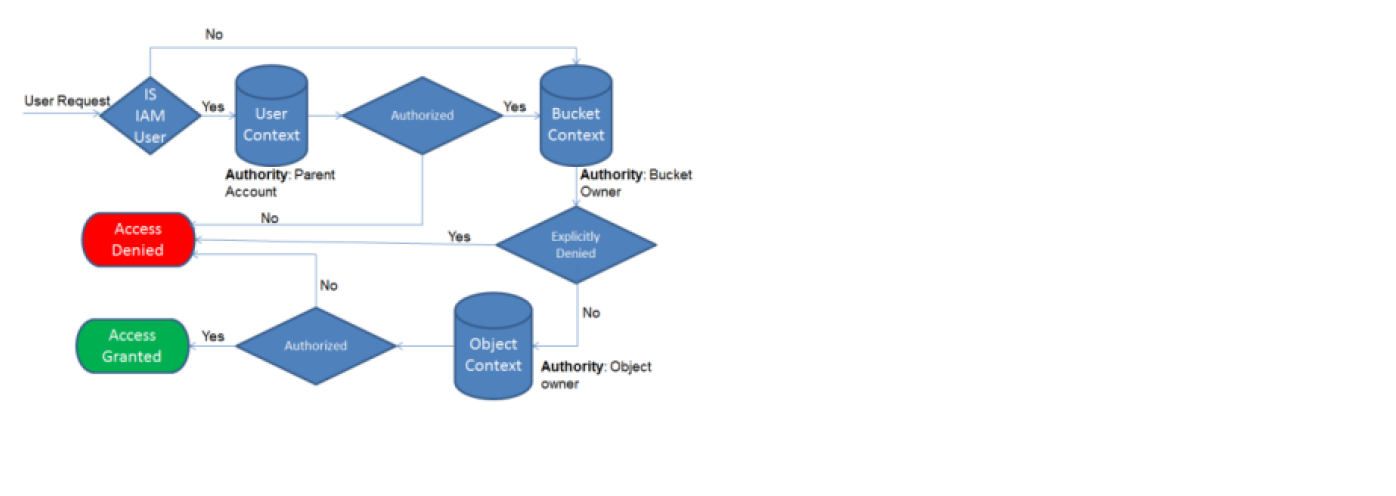
* Consider 3 Parents (AWS Account) A, B and C with Child (IAM User) A-User, B-User and C-User respectively
* Parent A owns a Toy box (Bucket) with Toy AAA and also allows toys (Objects) to be dropped and picked up
* Parent A can grant permission (User Policy OR Bucket policy OR both) to his Child A-User to access the Toy box and the toys
* Parent A can grant permissions (Bucket policy) to Parent B (different AWS account) to drop toys into the toys box. Parent B can grant permissions (User policy) to his Child B-User to drop Toy BAA
* Parent B can grant permissions (Object ACL) to Parent A to access Toy BAA
* Parent A can grant permissions (Bucket Policy) to Parent C to pick up the Toy AAA who in turn can grant permission (User Policy) to his Child C-User to access the toy
* Parent A can grant permission (through IAM Role) to Parent C to pick up the Toy BAA who in turn can grant permission (User Policy) to his Child C-User to access the toy

#### **Bucket Operation Authorization**



1. If the requester is an IAM user, the user must have permission (User Policy) from the parent AWS account to which it belongs
2. Amazon S3 evaluates a subset of policies owned by the parent account. This subset of policies includes the user policy that the parent account attaches to the user.
3. If the parent also owns the resource in the request (in this case, the bucket), Amazon S3 also evaluates the corresponding resource policies (bucket policy and bucket ACL) at the same time.
4. Requester must also have permissions (Bucket Policy or ACL) from the bucket owner to perform a specific bucket operation.
5. Amazon S3 evaluates a subset of policies owned by the AWS account that owns the bucket. The bucket owner can grant permission by using a bucket policy or bucket ACL.
6. Note that, if the AWS account that owns the bucket is also the parent account of an IAM user, then it can configure bucket permissions in a user policy or bucket policy or both

#### **Object Operation Authorization**

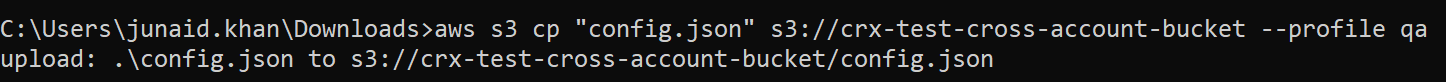


1. If the requester is an IAM user, the user must have permission (User Policy) from the parent AWS account to which it belongs.
2. Amazon S3 evaluates a subset of policies owned by the parent account. This subset of policies includes the user policy that the parent attaches to the user.
3. If the parent also owns the resource in the request (bucket, object), Amazon S3 evaluates the corresponding resource policies (bucket policy, bucket ACL, and object ACL) at the same time.
4. If the parent AWS account owns the resource (bucket or object), it can grant resource permissions to its IAM user by using either the user policy or the resource policy.
5. S3 evaluates policies owned by the AWS account that owns the bucket.
6. If the AWS account that owns the object in the request is not same as the bucket owner, in the bucket context Amazon S3 checks the policies if the bucket owner has explicitly denied access to the object.
7. If there is an explicit deny set on the object, Amazon S3 does not authorize the request.
8. Requester must have permissions from the object owner (Object ACL) to perform a specific object operation.
9. Amazon S3 evaluates the object ACL.
10. If bucket and object owners are the same, access to the object can be granted in the bucket policy, which is evaluated at the bucket context.
11. If the owners are different, the object owners must use an object ACL to grant permissions.
12. If the AWS account that owns the object is also the parent account to which the IAM user belongs, it can configure object permissions in a user policy, which is evaluated at the user context.

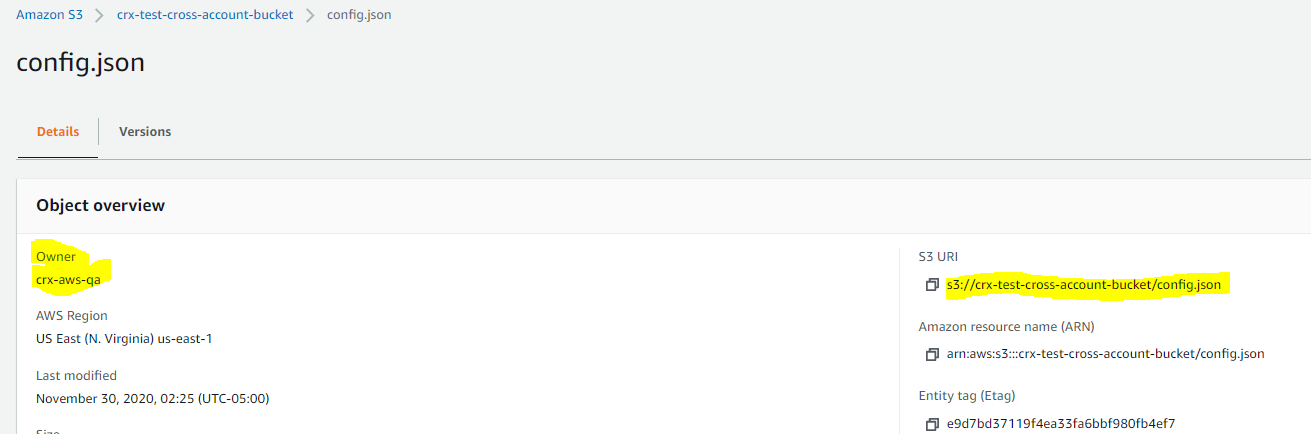
**Permission Delegation**

* If an AWS account owns a resource, it can grant those permissions to another AWS account.
* That account can then delegate those permissions, or a subset of them, to users in the account. This is referred to as permission delegation.
* But an account that receives permissions from another account cannot delegate permission cross-account to another AWS account.
* If the Bucket owner wants to grant permission to the Object which does not belong to it to an other AWS account it cannot do it through cross-account permissions and need to define a IAM role which can be assumed by the AWS account to gain access

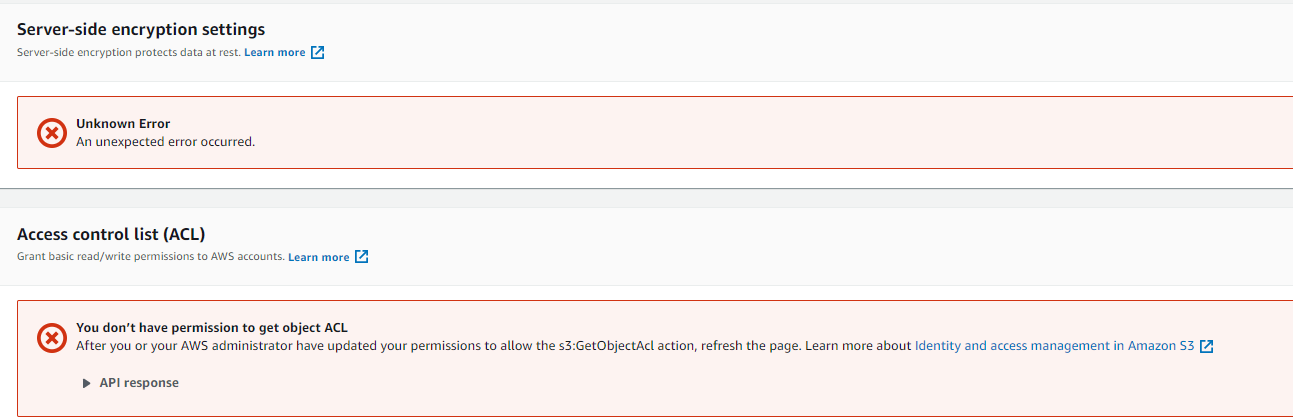
1. Use a user from QA account & Upload a file into development account bucket i.e. crx-test-cross-account-bucket

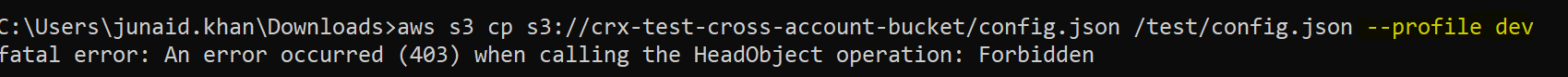


1. Since it was uploaded by a user from QA account, object ownership is with QA account.

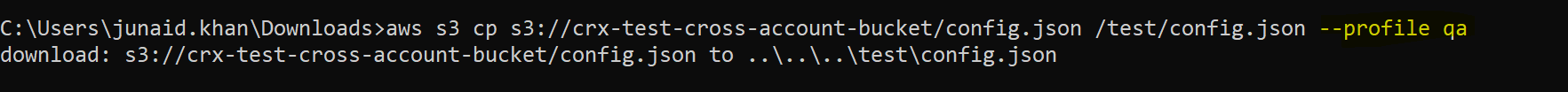


1. Although development account is bucket owner, users from development account won’t be able to access/download file uploaded by QA account.

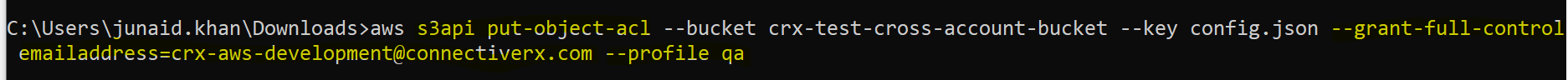




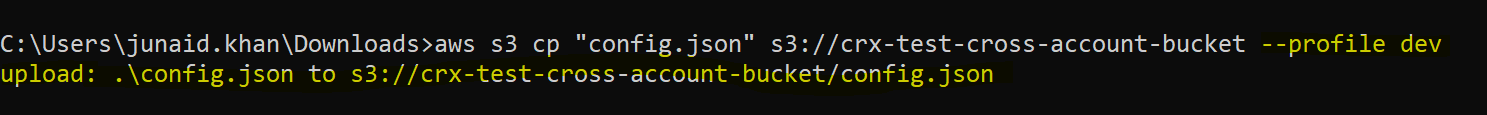
1. Although QA account is not bucket owner but users from QA account will be able to access/download file.

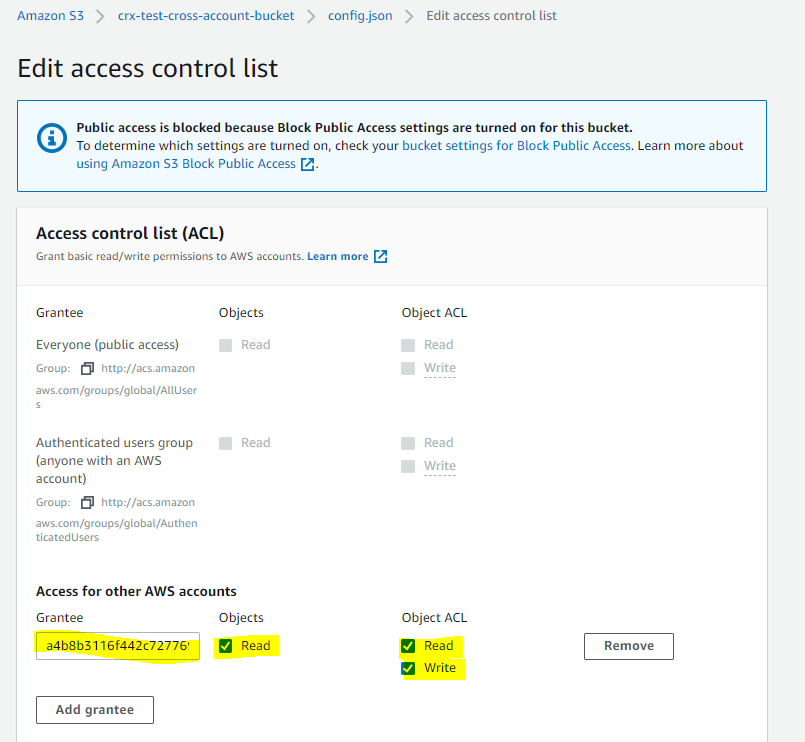


1. Since QA account is the object owner, it must grant access to development account(using canonical account id or email address) via object ACL.



1. Once QA account adds ACL for development account, users from development account will be able to access/download file uploaded by QA account.





## AWS Certification Exam Practice Questions

1. Which features can be used to restrict access to data in S3? Choose 2 answers
   1. **Set an S3 ACL on the bucket or the object.**
   2. Create a CloudFront distribution for the bucket.
   3. **Set an S3 bucket policy.**
   4. Enable IAM Identity Federation
   5. Use S3 Virtual Hosting
2. Which method can be used to prevent an IP address block from accessing public objects in an S3 bucket?
   1. **Create a bucket policy and apply it to the bucket**
   2. Create a NACL and attach it to the VPC of the bucket
   3. Create an ACL and apply it to all objects in the bucket
   4. Modify the IAM policies of any users that would access the bucket
3. A user has granted read/write permission of his S3 bucket using ACL. Which of the below mentioned options is a valid ID to grant permission to other AWS accounts (grantee. using ACL?
   1. IAM User ID
   2. S3 Secure ID
   3. Access ID
   4. **Canonical user ID**
4. A root account owner has given full access of his S3 bucket to one of the IAM users using the bucket ACL. When the IAM user logs in to the S3 console, which actions can he perform?
   1. He can just view the content of the bucket
   2. He can do all the operations on the bucket
   3. **It is not possible to give access to an IAM user using ACL**
   4. The IAM user can perform all operations on the bucket using only API/SDK
5. A root AWS account owner is trying to understand various options to set the permission to AWS S3. Which of the below mentioned options is not the right option to grant permission for S3?
   1. User Access Policy
   2. **S3 Object Policy**
   3. S3 Bucket Policy
   4. S3 ACL
6. A system admin is managing buckets, objects and folders with AWS S3. Which of the below mentioned statements is true and should be taken in consideration by the sysadmin?
   1. Folders support only ACL
   2. Both the object and bucket can have an Access Policy but folder cannot have policy
   3. Folders can have a policy
   4. **Both the object and bucket can have ACL but folders cannot have ACL**
7. A user has created an S3 bucket which is not publicly accessible. The bucket is having thirty objects which are also private. If the user wants to make the objects public, how can he configure this with minimal efforts?
   1. User should select all objects from the console and apply a single policy to mark them public
   2. User can write a program which programmatically makes all objects public using S3 SDK
   3. **Set the AWS bucket policy which marks all objects as public**
   4. Make the bucket ACL as public so it will also mark all objects as public
8. You need to configure an Amazon S3 bucket to serve static assets for your public-facing web application. Which methods ensure that all objects uploaded to the bucket are set to public read? Choose 2 answers
   1. **Set permissions on the object to public read during upload.**
   2. Configure the bucket ACL to set all objects to public read.
   3. **Configure the bucket policy to set all objects to public read.**
   4. Use AWS Identity and Access Management roles to set the bucket to public read.
   5. Amazon S3 objects default to public read, so no action is needed.
9. Amazon S3 doesn’t automatically give a user who creates \_\_\_\_\_ permission to perform other actions on that bucket or object.
   1. a file
   2. **a bucket or object**
   3. a bucket or file
   4. a object or file
10. A root account owner is trying to understand the S3 bucket ACL. Which of the below mentioned options cannot be used to grant ACL on the object using the authorized predefined group?
    1. Authenticated user group
    2. All users group
    3. Log Delivery Group
    4. **Canonical user group**
11. A user is enabling logging on a particular bucket. Which of the below mentioned options may be best suitable to allow access to the log bucket?
    1. Create an IAM policy and allow log access
    2. It is not possible to enable logging on the S3 bucket
    3. Create an IAM Role, which has access to the log bucket
    4. **Provide ACL for the logging group**
12. A user is trying to configure access with S3. Which of the following options is not possible to provide access to the S3 bucket / object?
    1. Define the policy for the IAM user
    2. Define the ACL for the object
    3. **Define the policy for the object**
    4. Define the policy for the bucket
13. A user is having access to objects of an S3 bucket, which is not owned by him. If he is trying to set the objects of that bucket public, which of the below mentioned options may be a right fit for this action?
    1. Make the bucket public with full access
    2. Define the policy for the bucket
    3. **Provide ACL on the object**
    4. Create an IAM user with permission
14. A bucket owner has allowed another account’s IAM users to upload or access objects in his bucket. The IAM user of Account A is trying to access an object created by the IAM user of account B. What will happen in this scenario?
    1. The bucket policy may not be created as S3 will give error due to conflict of Access Rights
    2. It is not possible to give permission to multiple IAM users
    3. **AWS S3 will verify proper rights given by the owner of Account A, the bucket owner as well as by the IAM user B to the object**
    4. It is not possible that the IAM user of one account accesses objects of the other IAM user