



## Configuring Amazon S3 security settings and access controls



Getting started at an AWS hosted workshop

### ▼ S3 Security Best Practices

#### ▼ Prepare Your Lab

Attach IAM Role to EC2 Instance

Connect to the EC2 Instance

Bucket Name

#### ▼ Lab 1 - S3 Security Exercises

Require HTTPS

Require SSE-KMS Encryption

Restrict Access to an S3 VPC Endpoint

Use AWS Config Rules to Detect a Public Bucket

Use Amazon Access Analyzer for S3

#### ▼ Lab 2 - S3 Access Grants

S3 Access Grants Lab - Initial Setup

Configure S3 Access Grants for IAM user

#### ▼ Lab 3 - Enabling Malware Protection for S3 by using GuardDuty

Enabling Malware Protection for S3 for your bucket

Testing GuardDuty Malware with an object.

#### ▼ Lab 4 - S3 Access Control Lists

Block Public ACLs

Configure S3 Block Public Access

**Disable S3 ACLs**

► Finding S3 access control lists with S3 Inventory

► Use Amazon Athena to query CloudTrail logs and identify S3 requests that depend on ACLs

Lab Summary

[Configuring Amazon S3 security settings and access controls](#) > [S3 Security Best Practices](#) > [Lab 4 - S3 Access Control](#)

## Disable S3 ACLs

In this exercise we will show you how to configure your buckets to disable ACLs and ensure objects are no longer granted access via any ACLs whether they be existing or new ACLs.

First, let us create an object with public read permissions via ACLs.

From your SSH session run the following command.

```
aws s3api put-object --key text01 --body textfile --acl public-read --bucket ${bucket}
```



```
ec2-user@storage-workshop ~]$ aws s3api put-object --key text01 --body textfile --acl public-read --bucket ${bucket}
{"SSEKMSKeyId": "arn:aws:kms:us-west-2:121049687582:key/7f76ab8c-d042-4392-afd8-020e04f9439a",
 "ETag": "\"c55218826e30a4264c101bd1e0344580\"",
 "ServerSideEncryption": "aws:kms"}
ec2-user@storage-workshop ~]$
```

This creates a text01 public object. Let's verify it is public with the following command.

```
aws s3api get-object-acl --key text01 --bucket ${bucket}
```



```
ec2-user@storage-workshop ~]$ aws s3api get-object-acl --key text01 --bucket ${bucket}
{"Owner": {
  "DisplayName": "ee-account+alb67042d295409c9da2f45a15d8865d",
  "ID": "8102705d191106b42514d17e31e40e53c140f258f758f4167cfb1a43aaff5972"
},
 "Grants": [
  {
    "Grantee": {
      "Type": "CanonicalUser",
      "DisplayName": "ee-account+alb67042d295409c9da2f45a15d8865d",
      "ID": "8102705d191106b42514d17e31e40e53c140f258f758f4167cfb1a43aaff5972"
    },
    "Permission": "FULL_CONTROL"
  },
  {
    "Grantee": {
      "Type": "Group",
      "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
    },
    "Permission": "READ"
  }
 ]
}
```

Notice that there is a Grantee Type Group with URI

`http://acs.amazonaws.com/groups/global/AllUsers` with Permission `READ`. This means public access is allowed to read this object.

We will update S3 Object Ownership to disable ACLs for all objects.

From the AWS console in the top search bar, search and select S3.

- Click the bucket name starting with `sid-security-xxxxxxx`.
- Click on the Permissions tab.
- Under Object Ownership click Edit.
- Select ACLs disabled (recommended).



- Click Save changes

## Edit Object Ownership Info

### Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**  
 All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**  
 Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

Cancel
Save changes

Return to your SSH session, and run the following command to verify that the object no longer shows that `http://acs.amazonaws.com/groups/global/AllUsers` has permission READ

```
aws s3api get-object-acl --key text01 --bucket ${bucket}
```



```
[ec2-user@storage-workshop ~]$ aws s3api get-object-acl --key text01 --bucket ${bucket}
{
  "Owner": {
    "DisplayName": "ee-account+alb67042d295409c9da2f45a15d8865d",
    "ID": "8102705d191106b42514d17e31e40e53c140f258f758f4167cfb1a43aaff5972"
  },
  "Grants": [
    {
      "Grantee": {
        "Type": "CanonicalUser",
        "DisplayName": "ee-account+alb67042d295409c9da2f45a15d8865d",
        "ID": "8102705d191106b42514d17e31e40e53c140f258f758f4167cfb1a43aaff5972"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

Without changing the ACL on the object, we have successfully blocked all public access to all objects within our bucket.

Let's see what happens if we try to create a new object with a public read ACL.

```
aws s3api put-object --key text01 --body textfile --acl public-read --bucket ${bucket}
```



The request should fail, notice how the error states that The bucket does not allow ACLs

```
[ec2-user@storage-workshop ~]$ aws s3api put-object --key text01 --body textfile --acl public-read --bucket ${bucket}
An error occurred (AccessControlListNotSupported) when calling the PutObject operation: The bucket does not allow ACLs
[ec2-user@storage-workshop ~]$
```

You have successfully disabled ACLs from this bucket and prevented new and existing objects from being granted permissions via ACLs.

[Previous](#)
[Next](#)
