



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

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

► Lab 3 - Enabling Malware Protection for S3 by using GuardDuty

► Lab 4 - S3 Access Control Lists

Lab Summary

[Configuring Amazon S3 security settings and access controls](#) > [S3 Security Best Practices](#) > [Lab 1 - S3 Security Exercises](#)

Require HTTPS

Bucket policies and user policies are two access policy options available for granting permission to your Amazon S3 resources. Both use JSON-based access policy language.

In this exercise, we will create a S3 Bucket Policy that requires connections to use HTTPS.

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 **Bucket Policy** click **Edit**.

Copy the bucket policy below, and paste into the Bucket Policy Editor.

```
{
  "Id": "S3-Security-Deny-unless-HTTPS",
  "Version": "2012-10-17",
  "Statement": [{
    "Action": "s3:*",
    "Effect": "Deny",
    "Principal": "*",
    "Resource": "arn:aws:s3::BUCKET_NAME/*",
    "Condition": {
      "Bool": {
        "aws:SecureTransport": false
      }
    }
  }]
}
```



Replace `BUCKET_NAME` with the bucket name you copied to your text editor.

Make sure you keep the `/*` at the end of the bucket name.

Your bucket policy will look similar to below.



Edit bucket policy [Info](#)

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.

[Policy examples](#)
[Policy generator](#)

Bucket ARN

 arn:aws:s3:::sid-security-900ed8e0-6a30-11ed-961c-0a7f8723e72d


Policy

```

1 {
2   "Id": "S3-Security-Deny-unless-HTTPS",
3   "Version": "2012-10-17",
4   "Statement": [{
5     "Action": "s3:*",
6     "Effect": "Deny",
7     "Principal": "*",
8     "Resource": "arn:aws:s3:::sid-security-900ed8e0-6a30-11ed-961c-0a7f8723e72d/*",
9     "Condition": {
10      "Bool": {
11        "aws:SecureTransport": false
12      }
13    }
14  }]
15 }
```

Click **Save** changes.

Open an SSH session to the SID-security-instance using EC2 Instance Connect if it is not already open. Run the following command.


```
aws s3api head-object --key app1/file1 --endpoint-url http://s3.amazonaws.com --bucket 
```

The command should return a 403 error since the endpoint-url is HTTP.

```

[ec2-user@storage-workshop ~]$
[ec2-user@storage-workshop ~]$ aws s3api head-object --key app1/file1 --endpoint-url http://s3.amazonaws.com --bucket ${bucket}
An error occurred (403) when calling the HeadObject operation: Forbidden
[ec2-user@storage-workshop ~]$
```

Now run the following command in your SSH session.

```
aws s3api head-object --key app1/file1 --endpoint-url https://s3.amazonaws.com --bucket 
```

The command succeeded because you used the s3api which uses HTTPS.

```

[ec2-user@storage-workshop ~]$
[ec2-user@storage-workshop ~]$ aws s3api --endpoint-url https://s3.amazonaws.com head-object --key app1/file1 --bucket ${bucket}
{
  "AcceptRanges": "bytes",
  "ContentType": "binary/octet-stream",
  "LastModified": "Sun, 08 Oct 2023 21:10:16 GMT",
  "ContentLength": 1048576,
  "ETag": "\"b6d81b360a5672d80c27430f39153e2c\"",
  "ServerSideEncryption": "AES256",
  "Metadata": {}
}
[ec2-user@storage-workshop ~]$
```

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