

## **Configuring Amazon** S3 security settings

Configuring Amazon S3 security settings and access controls > S3 Security Best Practices > Lab 1 - S3 Security Exercis

## 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
- ▶ Lab 3 Enabling Malware Protection for S3 by using GuardDutv
- ▶ Lab 4 S3 Access Control Lists Lab Summary

## 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-xxxxxxxx.
- 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
        }
   }]
}
```

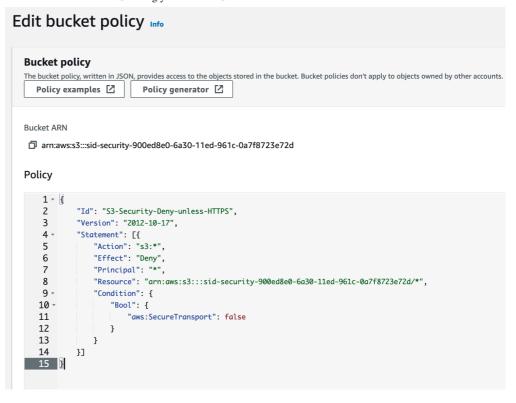


Replace BUCKET\_NAME with the bucket name you copied to your text editor.

① Make sure you keep the /\* at the end of the bucket name.

Your bucket policy will look similar to below.





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 (□):ke
```



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

```
ec2-user8storage-workshop -]$
ec2-user8storage-workshop -]$ aws s3api head-object --key app1/filel --endpoint-url http://s3.amazonaws.com --bucket ${bucket}
An error occurred (403) when calling the HeadObject operation: Forbidden [ec2-user@storage-workshop \neg]$
```

Now run the following command in your SSH session.

```
aws s3api head-object --key app1/file1 --endpoint-url https://s3.amazonaws.com --bucket(\boxed{\square})\tiny{ lcl}
```



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

```
serêstorage-workshop -]$
serêstorage-workshop -]$ aws s3api --endpoint-url https://s3.amazonaws.com head-object --key appl/filel --bucket ${bucket}
   ceptRanges": "bytes",
ntentType": "binary/octet-stream",
stModified: "Sun, 08 Oct 2023 21:10:16 GMT",
ntentLength: 104857.ag": "b6d81b360a5672d80c27430f39153e2c\"",
reveSideBruyption": "AES256",
tadata": {}
user@storage-workshop ~]$
```





