My Bucket, My Rules

**SPL-TF-200-SIBUCK-1 - Version 1.0.6**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

With more than 100 trillion objects in Amazon Simple Storage Service (Amazon S3) and an almost unimaginably broad set of use cases, securing data stored in Amazon S3 is important for every organization.

You’re the bucket owner and you want to ensure that the bucket and its contents are compliant with the security guidelines and compliance regulations of your organization. This lab demonstrate some examples of Amazon S3 preventative security best practices. Its goal is to provide you with the skills that you need to successfully configure and test policies to enforce the following:

* Where the bucket is accessed from.
* Access permissions.
* Encryption at rest and in transit.
* The type of encryption that is required for compliance.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Configure the bucket policy to enforce HTTPS connections only.
* Configure the bucket policy to accept connections only through the virtual private cloud (VPC) endpoint.
* Configure bucket policy to only accept object uploads that use an accepted encryption method and encryption key.
* Test these requirements using the AWS Command Line Interface (AWS CLI).

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with the following services or features:

* Amazon S3
* AWS Identity and Access Management (IAM)
* Amazon Virtual Private Cloud (Amazon VPC)
* VPC gateway endpoints
* AWS Key Management Service (AWS KMS)

DURATION

This lab requires approximately *60* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command to run from the terminal.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Learn more:** Where to find more information.
* **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
* **CAUTION:** Information of special interest or importance (not so important as to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Hint:** A hint to a question or challenge.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

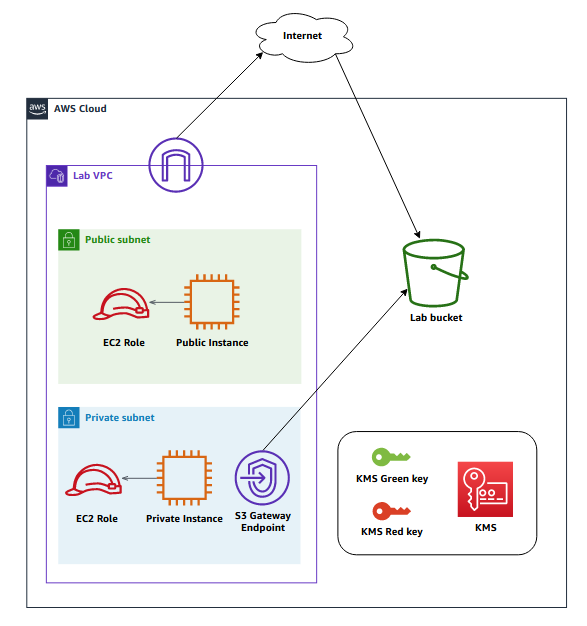
**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

LAB ENVIRONMENT

The following diagram shows the basic architecture of the lab environment:



*Architecture diagram of the lab environment. More details in the image description.*

**Image description:** The following list details the major resources in the diagram:

* An *S3* bucket.
* A *VPC* with one public subnet and one private subnet.
  + The public subnet has direct access to the internet through the internet gateway.
  + The private subnet has access to Amazon S3 through a VPC gateway endpoint.
* An *EC2* instance in each subnet.
  + The EC2 instances have permissions to list, put, and get objects from the S3 bucket through an Amazon EC2 IAM role.
* Two AWS KMS keys named as *Green Key* and *Red Key*.

**Note:** The pane to the left of these instructions includes a list of all the lab resources that you need to complete the lab activities.

**Task 1: Testing Amazon S3 connectivity and uploading test objects**

In this task, you perform the following:

* Connect to both the public and private instances through Session Manager, a capability of AWS Systems Manager.
* Set Linux variables on both instances that you use in your AWS CLI commands.
* Run some AWS CLI commands to verify access to the lab bucket.
* Upload a test object to the bucket.
* Check the current bucket policy settings on the lab bucket.

**Hint:** Throughout the lab, you use the terminal to run AWS CLI commands on both public and private instances. Make sure that you open each terminal on a separate browser tab. Also, before you run a command, double-check that you are using the correct terminal (whether for the public or private instance).

1. To connect to the *Public-Instance* terminal, copy the **PublicInstanceSSMUrl** value that is listed to the left of these instructions. Paste the URL into a new web browser tab and press **Enter**. This opens the *Public-Instance* terminal with the *Public-Instance$* shell prompt.
2. **Command:** To create an object on the **Public-Instance**, run the following command:

echo 'This is the 1st test object for the lab' > object01.txt

**Expected output:** None, unless there is an error.

In this lab, you run many commands from the command line interface (CLI) after replacing some parameters in the sample command, such as the bucket name and other attributes. You set some variables in this section, which simplifies copying the commands into your Amazon EC2 terminal.

1. **Command:** To set Linux shell variables on the **Public-Instance**, run the following commands after making these changes:

* Replace the **INSERT\_LAB\_BUCKET\_NAME** placeholder value with the **LabBucketName** value that is listed to the left of these instructions.
* Replace the **INSERT\_LAB\_REGION\_CODE** placeholder value with the **AwsRegionName** value that is listed to the left of these instructions.

**Copy edit:** You might want to copy these commands to your preferred text editor, update the values, and then paste them back into the terminal.

lab\_bucket=INSERT\_LAB\_BUCKET\_NAME

lab\_region=INSERT\_LAB\_REGION\_CODE

**Expected output:** None, unless there is an error.

1. **Command:** To verify that the variables are set correctly, run the following commands on the **Public-Instance**, and validate the output:

echo $lab\_bucket

echo $lab\_region

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Public-Instance$ echo $lab\_bucket

mylabbucket-random\_number

Public-Instance$ echo $lab\_region

us-west-2

**Note:**

* If your session terminates, you need to set these variables again.
* In each task, we show the command using the variables you set so that you can directly copy them into your terminal. The command should work if the variables are set correctly.

1. **Command:** To upload the object that you created into the bucket, run the following command on the **Public-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body object01.txt \

--key object01.txt

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"SSEKMSKeyId": "arn:aws:kms:us-west-2:111122223333:key/1c446a23-24aa-4388-8f0f-0ff2f5b37c86",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"ServerSideEncryption": "aws:kms"

}

The output indicates that the upload was successful.

1. **Command:** To list the objects in the bucket, run the following command on the **Public-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

You should be able to see the contents in the bucket.

1. To connect to the *Private-Instance* terminal, copy the **PrivateInstanceSSMUrl** value that is listed to the left of these instructions. Paste the URL into a new web browser tab and press **Enter**. This opens the *Private-Instance* terminal with the *Private-Instance$* shell prompt.

**Note:** Make sure that you differentiate between the terminals of the public and private instance and that you enter the commands in the correct terminal.

1. **Command:** To set Linux shell variables on the **Private-Instance**, run the following commands after making these changes:

* Replace the **INSERT\_LAB\_BUCKET\_NAME** placeholder value with the **LabBucketName** value that is listed to the left of these instructions.
* Replace the **INSERT\_LAB\_REGION\_CODE** placeholder value with the **AwsRegionName** value that is listed to the left of these instructions.
* Replace the **INSERT\_KMS\_GREEN\_KEY\_ID** placeholder value with the **KMSGreenKeyID** value that is listed to the left of these instructions.
* Replace the **INSERT\_KMS\_RED\_KEY\_ID** placeholder value with the **KMSRedKeyID** value that is listed to the left of these instructions.

**Copy edit:** You might want to copy these commands to your preferred text editor, update the values, and then paste them back into the terminal.

lab\_bucket=INSERT\_LAB\_BUCKET\_NAME

lab\_region=INSERT\_LAB\_REGION\_CODE

kms\_green\_key\_id=INSERT\_KMS\_GREEN\_KEY\_ID

kms\_red\_key\_id=INSERT\_KMS\_RED\_KEY\_ID

**Expected output:** None, unless there is an error.

1. **Command:** To verify that the variables are set correctly, run the following commands on the **Private-Instance** and validate the output:

echo $lab\_bucket

echo $lab\_region

echo $kms\_green\_key\_id

echo $kms\_red\_key\_id

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private-Instance$ echo $lab\_bucket

mylabbucket-random\_number

Private-Instance$ echo $lab\_region

us-west-2

Private-Instance$ echo $kms\_green\_key\_id

1c446a23-24aa-4388-8f0f-0ff2f5b37c86

Private-Instance$ echo $kms\_red\_key\_id

30892466-ed5b-4f32-9a0d-fd8e344421fd

**Note:**

* If your session terminates, you need to set these variables again.
* In each task, we show the command using the variables that you set, so you can directly copy them into your terminal. The command should work if the variables are set correctly.

1. **Command:** To list the objects in the bucket, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

You should be able to see the contents in the bucket. This indicates that you can connect to the bucket through the gateway VPC endpoint, which was created as part of the pre-lab build.

1. At the top of the AWS Management Console, in the search bar, search for and choose

S3

.

1. Under **Buckets** section, choose the bucket link which has **mylabbucket** string in its name.
2. Choose the **Permissions** tab, scroll down to the **Bucket Policy** section and check the current bucket policy.

The policy is empty. This means that all permissions are now controlled through the IAM policy on the identity that makes the API calls (EC2 instances in this scenario).

In the next few tasks of this lab, you start configuring your bucket policy to enforce certain requirements for accessing your bucket.

**Congratulations!** You verified that you can connect to the lab bucket from both instances, set Linux variables, uploaded a test object, and verified that the bucket has no bucket policy applied to it.

**Task 2: Enforcing HTTPS connections**

In this task, you perform the following:

* Test accessing the bucket using HTTP instead of HTTPS.
* Configure a bucket policy to restrict access to the bucket using only HTTPS and verify it.

In the previous task, you used the HTTPS endpoint when making the API calls from the AWS CLI. So, the calls were made using HTTPS on TCP port 443. If the endpoint supports HTTP, you can use HTTP calls by choosing the protocol in the request.

1. **Command:** To test if you can make HTTP calls to the bucket from the **Public-Instance**, run the following command:

aws --endpoint-url http://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

You should be able to see the contents in the bucket.

1. **Command:** To test if you can make HTTP calls to the bucket from the **Private-Instance**, run the following command:

aws --endpoint-url http://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

You should be able to see the contents in the bucket.

Both instances should be able to use HTTP calls as there is no HTTPS enforcement configured yet.

**Learn more:** Some AWS services do not support HTTP. Refer to *Service Endpoints and Quotas* in the **Additional resources** section for more information.

For compliance reasons, you need to ensure that only HTTPS connections are allowed to the bucket (encryption in transit). As the bucket owner, you can only control your bucket policy and have no control over the identity-based policies attached to instances or applications.

**Do it yourself**

1. Configure a bucket policy on your bucket to deny any connections that do not use HTTPS. The bucket policy must deny all Amazon S3 actions to the bucket and its objects from any principal if HTTP protocol is used. You need to find a suitable IAM condition that helps you achieve your goal.

**Hint:** For guidance on how to complete this task, see [What S3 Bucket Policy Should I Use to Comply with the AWS Config Rule S3-bucket-ssl-requests-only?](https://aws.amazon.com/premiumsupport/knowledge-center/s3-bucket-policy-for-config-rule/).

**If you need help, expand this section for the full solution:**

TASK 2 VERIFICATION

Now, verify that the bucket policy that you configured is correct.

1. **Command:** To test if you can make HTTPS calls to the bucket from the **Public-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

This should be successful, and you can view the contents in the bucket because the call was made using HTTPS.

1. **Command:** To test if you can make HTTP calls to the bucket from the **Public-Instance**, run the following command:

aws --endpoint-url http://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the ListObjects operation: Access Denied

This should be denied if your bucket policy is correct because it used HTTP.

1. **Command:** To test if you can make HTTPS calls to the bucket from the **Private-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

This should be successful, and you can view the contents in the bucket because the call was made using HTTPS.

1. **Command:** To test if you can make HTTP calls to the bucket from the **Private-Instance**, run the following command:

aws --endpoint-url http://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the ListObjects operation: Access Denied

This should be denied if your bucket policy is correct because it used HTTP.

**Congratulations!** You have applied the first rule of your compliance requirements on your bucket which restricts access to the bucket to HTTPS only and validated it.

**Task 3: Enforcing access to the bucket through the VPC endpoint**

In this task, you perform the following:

* Add a second statement to your bucket policy to restrict bucket access to be only through the private subnet and specifically, the VPC gateway endpoint.
* Validate that the bucket policy is correct.

At the moment, the S3 bucket is accessed from both the public and private instances using different paths:

* The public instance is accessing Amazon S3 through the internet gateway.
* The private instance is accessing the Amazon S3 AWS network, without the need for internet or network address translation (NAT) gateways.

Your lab setup already has a configured VPC endpoint (gateway endpoint) in the private subnet, which provided the private instance with access to the bucket in the previous tasks.

**Learn more:** Refer to *Gateway Endpoints* in the **Additional resources** section for more information.

Based on compliance requirements, you want to restrict the access to the bucket to be only through the private subnet and specifically, through the VPC gateway endpoint. Because you don’t have access or control over the VPC configuration, you need to enforce this requirement on your bucket using the bucket policy.

**Do it yourself**

1. Modify your bucket policy by adding a new statement that denies any connections that do not originate in the Amazon S3 gateway endpoint in the private subnet. You can find the Amazon S3 gateway endpoint VPC endpoint ID in the pane to the left of these instructions. You need to find a suitable IAM condition that helps you achieve your goal. Read the following **CAUTION** before you configure the policy:

**CAUTION: Follow these guidelines in your bucket policy statement:**

* Ensure that you only include these **Actions** in your bucket policy statement:
  + s3:GetObject
  + s3:PutObject
  + s3:ListBucket
* Do not use a wildcard “\*” for the actions because you might lock yourself completely out of the bucket.
* For this specific statement, use the Amazon EC2 role Amazon Resource Name (ARN) value as the principal instead of the wildcard principal “\*”. By doing so, you can continue browsing the bucket objects from the console and checking their properties, which you need later in the lab. You can find the Amazon EC2 role ARN in the pane to the left of these instructions.

**Learn more:** Refer to *IAM role principals - AWS JSON Policy Elements: Principal* in the **Additional resources** section for more information.

**Hint:** For guidance on how to complete this task, see [Controlling Access from VPC Endpoints with Bucket Policies](https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies-vpc-endpoint.html).

**If you need help, expand this section for the full solution:**

TASK 3 VERIFICATION

Now, verify that the bucket policy that you configured is correct.

**Testing from Public-Instance**

1. **Command:** To test if you can list the bucket objects from the **Public-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the ListObjects operation: Access Denied

1. **Command:** To test if you can put an object to the bucket from the **Public-Instance** (by uploading the same file but with a new object name), run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body object01.txt \

--key new\_object01.txt

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

1. **Command:** To test if you can get an object from the **Public-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api get-object \

--bucket $lab\_bucket \

--key object01.txt \

new\_object01.txt

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the GetObject operation: Access Denied

All testing from the public instance should be denied because it is not originating from the VPC endpoint in the private subnet.

**Testing from Private-Instance**

1. **Command:** To test if you can list the bucket objects from the **Private-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api list-objects --bucket $lab\_bucket

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"Contents": [

{

"LastModified": "2022-09-12T02:20:00.000Z",

"ETag": "\"f9a29703c427f46f94fd76e5baf2222f\"",

"StorageClass": "STANDARD",

"Key": "object01.txt",

"Size": 40

}

]

}

1. **Command:** To create an object on the **Private-Instance**, run the following command:

echo 'This is the 2nd test object for the lab' > object02.txt

**Expected output:** None, unless there is an error.

1. **Command:** To test if you can put an object to the bucket from the **Private-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body object02.txt \

--key object02.txt

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"SSEKMSKeyId": "arn:aws:kms:us-west-2:111122223333:key/1c446a23-24aa-4388-8f0f-0ff2f5b37c86",

"ETag": "\"4b3eb9639d00b7c6e4f6ab26d4af785f\"",

"ServerSideEncryption": "aws:kms"

}

1. **Command:** To test if you can get an object from the **Private-Instance**, run the following command:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api get-object \

--bucket $lab\_bucket \

--key object02.txt \

new\_object02.txt

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"AcceptRanges": "bytes",

"ContentType": "binary/octet-stream",

"LastModified": "Mon, 12 Sep 2022 02:55:26 GMT",

"ContentLength": 40,

"ETag": "\"4b3eb9639d00b7c6e4f6ab26d4af785f\"",

"ServerSideEncryption": "aws:kms",

"SSEKMSKeyId": "arn:aws:kms:us-west-2:111122223333:key/1c446a23-24aa-4388-8f0f-0ff2f5b37c86",

"Metadata": {}

}

All testing from the private instance should be successful because it is originating from the VPC endpoint.

1. Close the **Public-Instance** Session Manager window because you do not need it after this task.

**Congratulations!** You have applied the second rule of your compliance requirements on your bucket which restricts access to the bucket only from the private subnet via the gateway endpoint and validated it.

**Task 4: Restricting object uploads to your preferred encryption option and AWS KMS key**

In this task, you perform the following:

* Test uploading objects to the bucket using different encryption options and keys.
* Add a third statement to your bucket policy to enforce the AWS KMS encryption option and your preferred AWS KMS key for all uploads to the bucket.
* Validate that the bucket policy is correct.

The lab environment has two AWS KMS keys created. The aliases of these two keys are as follows:

* kms-green-key
* kms-red-key

The IDs of both keys are provided in the pane to the left of these instructions.

1. In the **Amazon S3** page, under **Buckets** section, choose the bucket link which has **mylabbucket** string in its name.
2. Choose the **Properties** tab and scroll down to check the **Default Encryption** settings on the bucket. Notice that the bucket is set to use SSE-KMS using a specific AWS KMS key, which matches the ID of kms-green-key.

When you uploaded the previous objects, you did not specify any encryption option or key to be used.

1. In the bucket page, choose the **Objects** tab, then choose the object link of any object that you uploaded in the previous tasks.
2. In the object page, scroll down to the **Server-side encryption settings** section. Notice that this object used kms-green-key as the main key for envelop encryption because it is set as the default key for the bucket. This is the only key that you approve in your bucket as part of the compliance.

**Note:** All the remaining testing now occurs from the private instance because it is the only instance that you can access the bucket with.

**Learn more:** Refer to *Protecting Data Using Server-Side Encryption* in the **Additional resources** section for more information.

1. **Command:** To create another object on the **Private-Instance**, run the following command:

echo 'This is a red key object' > red\_object.txt

**Expected output:** None, unless there is an error.

1. **Command:** To upload the new object using kms-red-key, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body red\_object.txt \

--key red\_object.txt \

--server-side-encryption aws:kms \

--ssekms-key-id $kms\_red\_key\_id

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"SSEKMSKeyId": "arn:aws:kms:us-west-2:111122223333:key/30892466-ed5b-4f32-9a0d-fd8e344421fd",

"ETag": "\"c868e21b537a10f79f4670013677890e\"",

"ServerSideEncryption": "aws:kms"

}

The upload should be successful.

1. In the **Objects** tab under your bucket, choose the **red\_object.txt** link.
2. In the object page, scroll down to the **Server-side encryption settings** section. Notice that this object used the kms-red-key as the main key for envelop encryption.
3. **Command:** To create another object on the **Private-Instance**, run the following command:

echo 'This is an sse-s3 object' > sses3\_object.txt

**Expected output:** None, unless there is an error.

1. **Command:** To upload the new object using SSE-S3, which does not even use your AWS KMS keys, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body sses3\_object.txt \

--key sses3\_object.txt \

--server-side-encryption AES256

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"ETag": "\"11eba73b35d5f3120128614016e52d22\"",

"ServerSideEncryption": "AES256"

}

The upload should be successful.

1. On the **Objects** tab, under your bucket, choose **sses3\_object.txt** link.
2. In the object page, scroll down to the **Server-side encryption settings** section. Notice that this object used SSE-S3.

What you can conclude from these tests is that setting the bucket default encryption will only be applied if you don’t specify any encryption or key options during uploads. However, you can override the default settings, if you choose to, during uploads. As a bucket owner, you can see that there might be a compliance issue, and you want to ensure that all uploads use AWS KMS and your preferred key, which is kms-green-key in this scenario.

**Do it yourself**

1. Modify your bucket policy by adding a new statement that denies any object uploads that do not use kms-green-key.

**Hint:** For guidance on how to complete this task, see [Protecting Data Using Server-Side Encryption with AWS Key Management Service (SSE-KMS)](https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingKMSEncryption.html).

**Note:** For the purpose of this lab scenario, when you use the above reference, consider the following:

* You do not need the bucket policy statement with the Null condition because the bucket is already configured with default encryption. However, if you decide to use it, it still meets the requirements.
* Because you want to enforce using a specific AWS KMS key, consider using the following IAM condition: s3:x-amz-server-side-encryption-aws-kms-key-id to specify the key ID instead of the encryption method.
* With this condition, you need to use the key ARN and not the key ID in the policy.
* You only need the s3:PutObject action in this statement.

**If you need help, expand this section for the full solution:**

TASK 4 VERIFICATION

Now, verify that the bucket policy that you configured is correct.

1. **Command:** To test if you can upload an object using kms-red-key, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body red\_object.txt \

--key red\_object.txt \

--server-side-encryption aws:kms \

--ssekms-key-id $kms\_red\_key\_id

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

1. **Command:** To test if you can upload an object using SSE-S3, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body sses3\_object.txt \

--key sses3\_object.txt \

--server-side-encryption AES256

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

If your bucket policy is correct, both of these tests that you just performed should result in denials because they do not use the required encryption option and key.

1. **Command:** To test if you can upload an object using kms-green-key, run the following command on the **Private-Instance**:

aws --endpoint-url https://s3.$lab\_region.amazonaws.com s3api put-object \

--bucket $lab\_bucket \

--body object02.txt \

--key task4\_object.txt \

--server-side-encryption aws:kms \

--ssekms-key-id $kms\_green\_key\_id

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"SSEKMSKeyId": "arn:aws:kms:us-west-2:111122223333:key/1c446a23-24aa-4388-8f0f-0ff2f5b37c86",

"ETag": "\"e36932ab26479f7f056817b550637930\"",

"ServerSideEncryption": "aws:kms"

}

This test should be successful, and you can verify that it is by using the correct key on the object’s properties.

The lab scenario might not exactly match your environment, but you should now have the skills required to enforce your rules on your buckets.

**Learn more:** Refer to *Security Best Practices for Amazon S3* in the **Additional resources** section for more information.

**Congratulations!** You have applied the third rule of your compliance requirements on your bucket to enforce uploading objects using only a specific encryption option and encryption key, and validated it.

**Summary**

**Congratulations!** You have now successfully done the following:

* Configured your bucket policy to only accept HTTPS connections.
* Configured your bucket policy to only accept API calls that originate from a specific subnet through the VPC endpoint.
* Configured your bucket policy to only accept uploads that use a specific encryption method and AWS KMS key.
* Applied your rules to your bucket based on your compliance requirements.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Service Endpoints and Quotas](https://docs.aws.amazon.com/general/latest/gr/aws-service-information.html).
* [Gateway Endpoints](https://docs.aws.amazon.com/vpc/latest/privatelink/gateway-endpoints.html).
* [IAM role principals - AWS JSON Policy Elements: Principal](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html#principal-roles).
* [Protecting Data Using Server-Side Encryption](https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html).
* [Security Best Practices for Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html).
* [Amazon S3 documentation](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html).
* [Amazon VPC documentation](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html).
* [IAM documentation](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html).
* [AWS KMS documentation](https://docs.aws.amazon.com/kms/latest/developerguide/overview.html).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).