30 DAYS CLOUD SOLUTIONS DEVELOPMENT CHALLENGE

DAY 1

LAB PROJECT 1
S3 POLICY ENFORCEMENT

Description:
This lab focuses on enforcing a strong security baseline on an Amazon S3 bucket using its Bucket Policy. The primary goal is to use a policy to mandate that all objects uploaded to the bucket must be encrypted using AWS Key Management Service (SSE-KMS). This security measure prevents unencrypted or weakly encrypted data from ever residing in the bucket, fulfilling a common compliance requirement.

Objective: To configure and validate an S3 Bucket Policy that uses an explicit `Deny` condition to mandate Server-Side Encryption with AWS Key Management Service (SSE-KMS) for all `s3:PutObject` operations.

The lab involves setting up three core components:

1. An AWS KMS Key for the encryption.
2. An S3 Bucket Policy with an explicit `Deny` statement if the upload request is missing the required KMS headers.
3. A restricted IAM User with the *identity-based* permission to upload files, but without the *knowledge* or *ability* to satisfy the KMS encryption condition, demonstrating the policy's effectiveness.

The outcome will be a successful access attempt (when the required encryption header is correctly specified) and a failed access attempt (when the header is missing), proving that the Bucket Policy (Resource-based policy) overrides the IAM User's allowed permissions unless all conditions are met.

Security PrincipleDefense in Depth and Policy Precedence (Resource Policies overriding Identity Policies unless conditions are met).

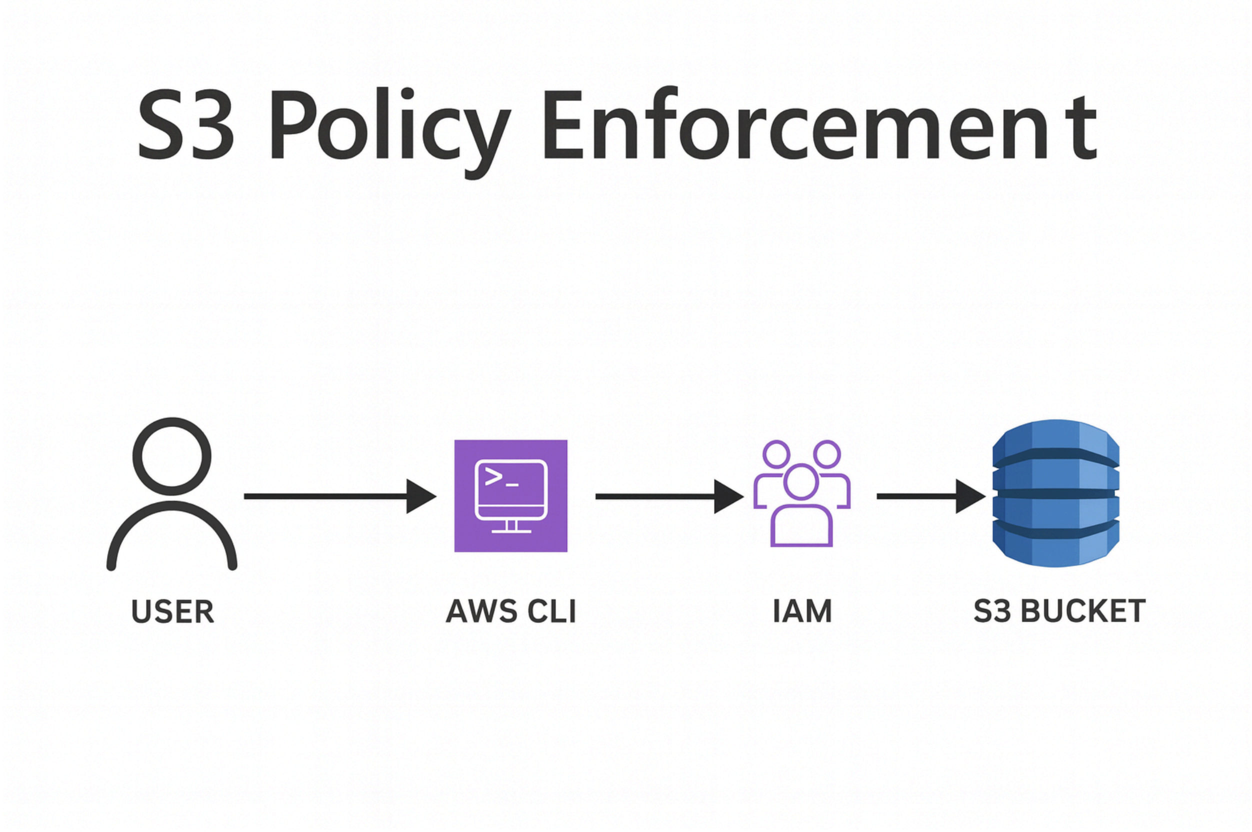
---

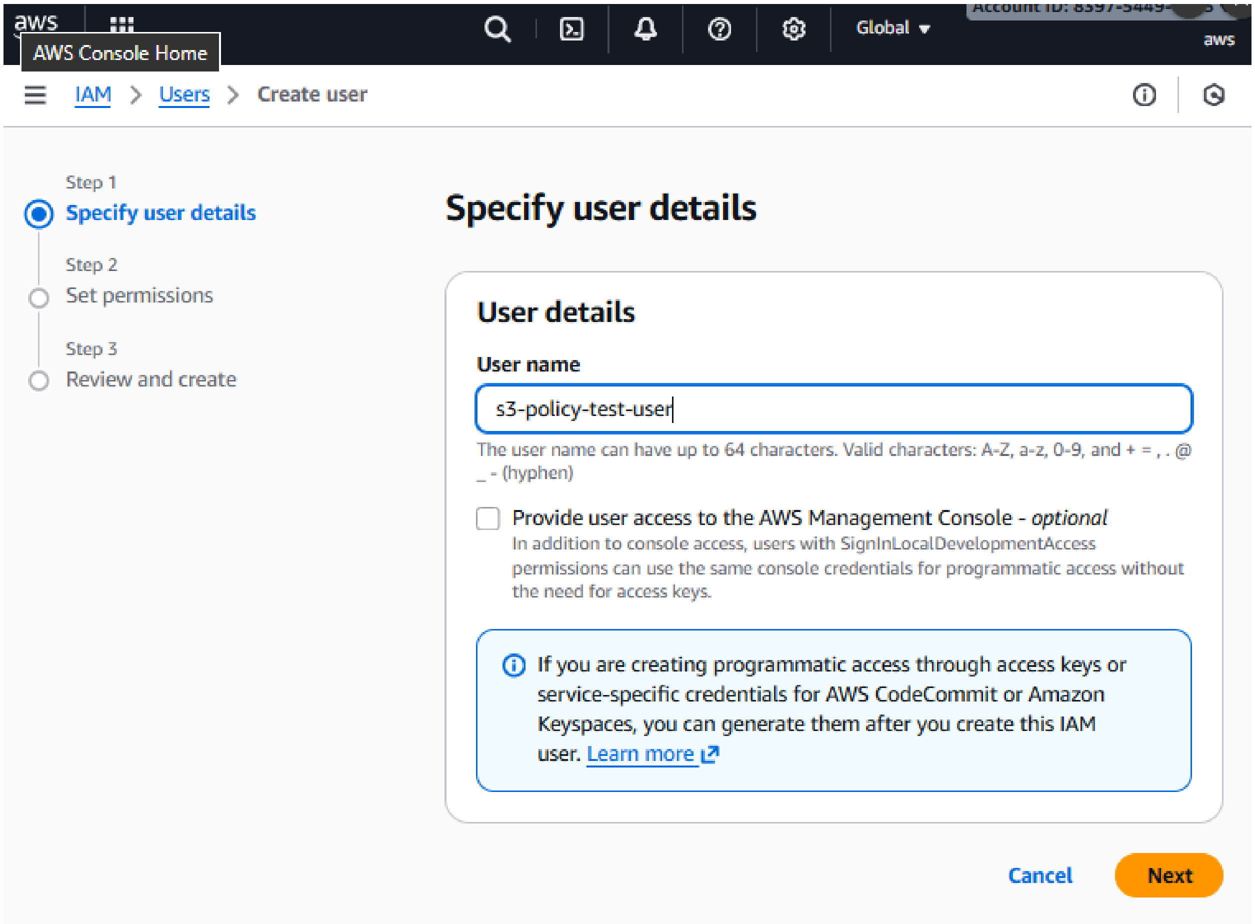
*Architecture and Prerequisites

The lab demonstrates the interaction between an Identity Policy (IAM User) that grants permission and a Resource Policy (Bucket Policy) that sets a mandatory condition. The upload is successful *only* when the condition is satisfied.

Prerequisites

- Active AWS Account with administrative access.
- AWS CLI installed and configured.
- A basic text file (`test-file.txt`) for the upload attempts.
- The AWS Region (e.g., `af-south-1`) is known and consistently used.





Step 1
● Specify user details

Step 2
○ Set permissions

Step 3
○ Review and create

Specify user details

User details

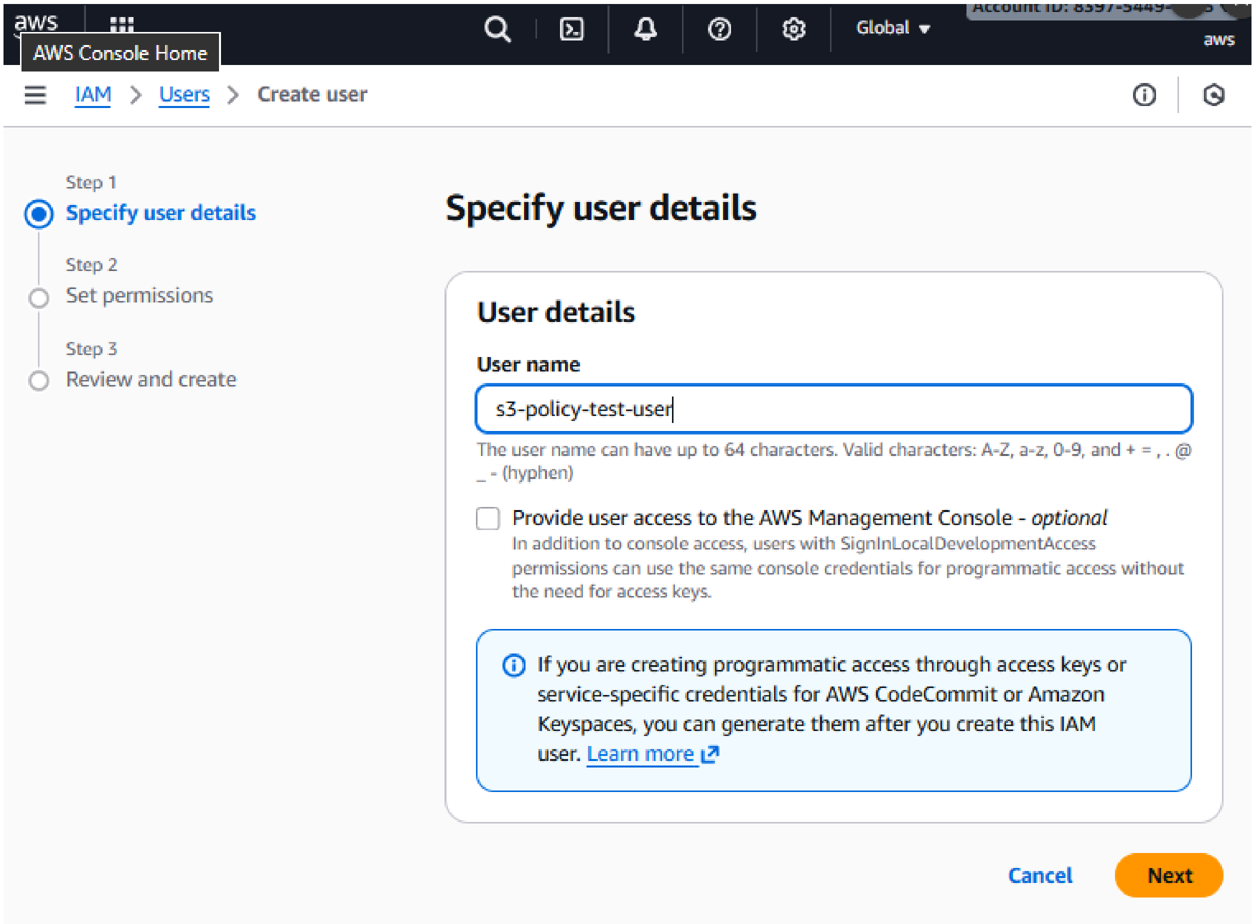
User name

s3-policy-test-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ Provide user access to the AWS Management Console - *optional*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more



Account ID: 8397-5449-[redacted]

IAM > Users > Create user

Step 1
● Specify user details

Step 2
○ Set permissions

Step 3
○ Review and create

Specify user details

User details

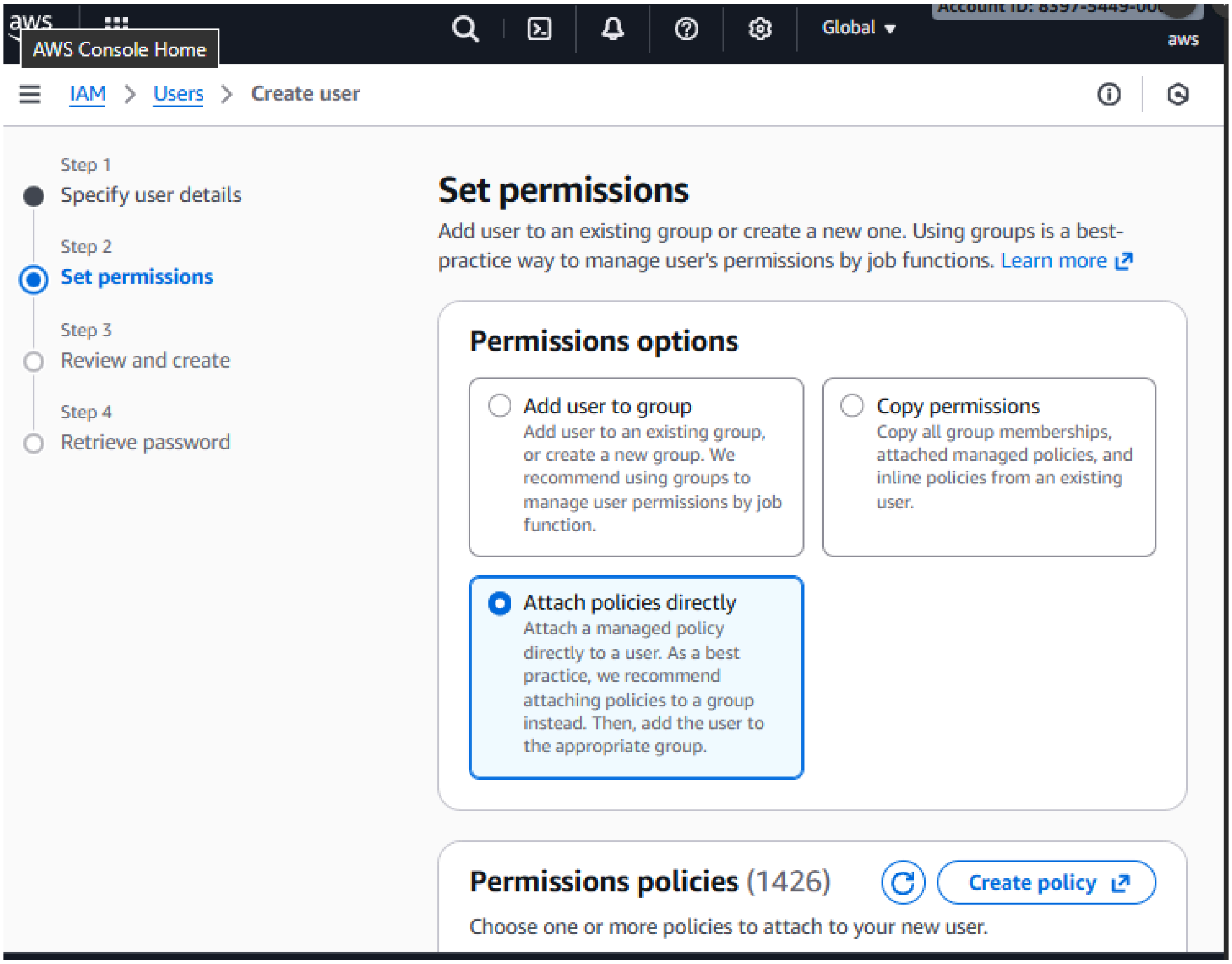
User name

s3-policy-test-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ Provide user access to the AWS Management Console - *optional*
In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more



Step 1
● Specify user details

Step 2
● Set permissions

Step 3
○ Review and create

Step 4
○ Retrieve password

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

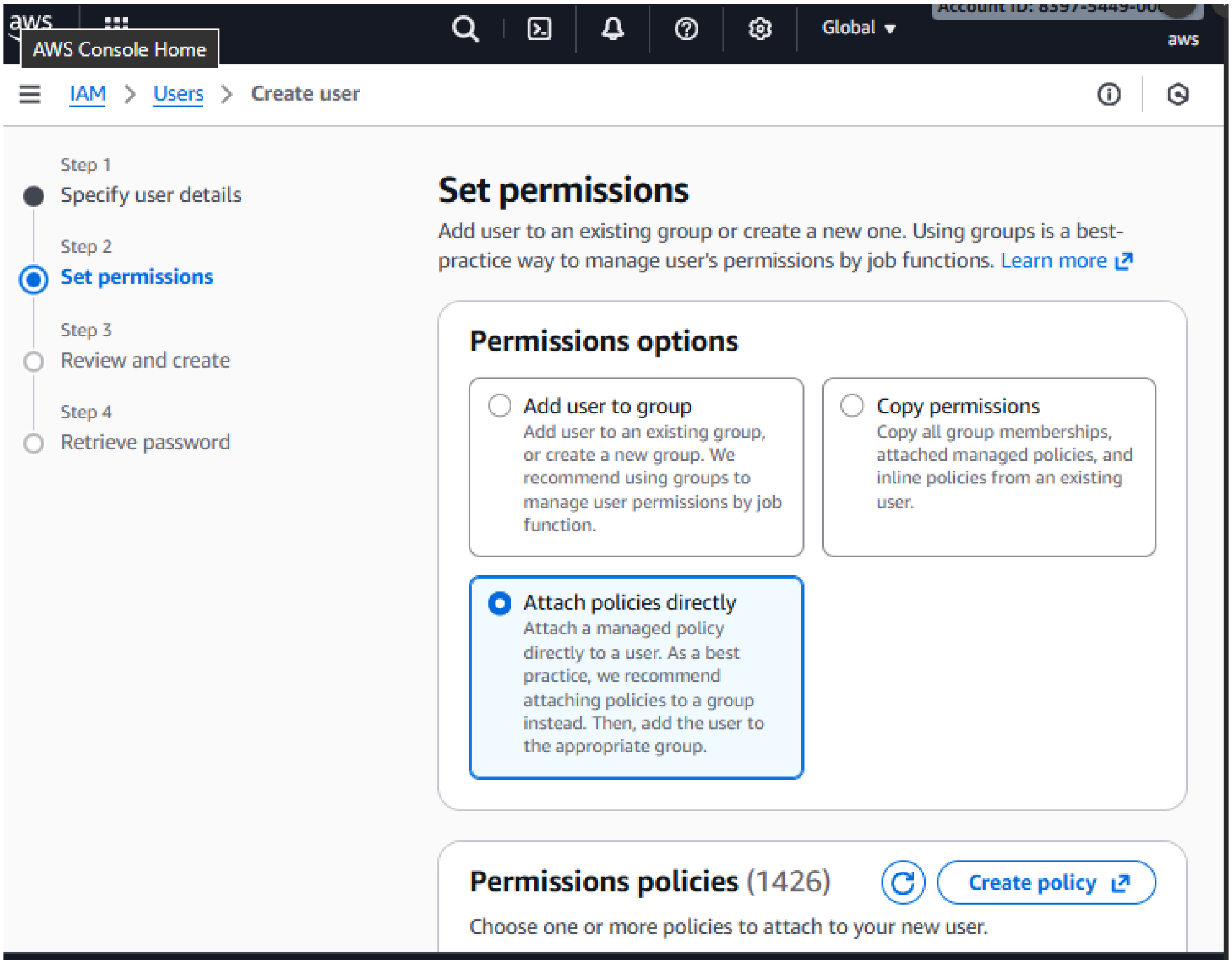
○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1426)

Choose one or more policies to attach to your new user.



Step 1
● Specify user details

Step 2
● Set permissions

Step 3
○ Review and create

Step 4
○ Retrieve password

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

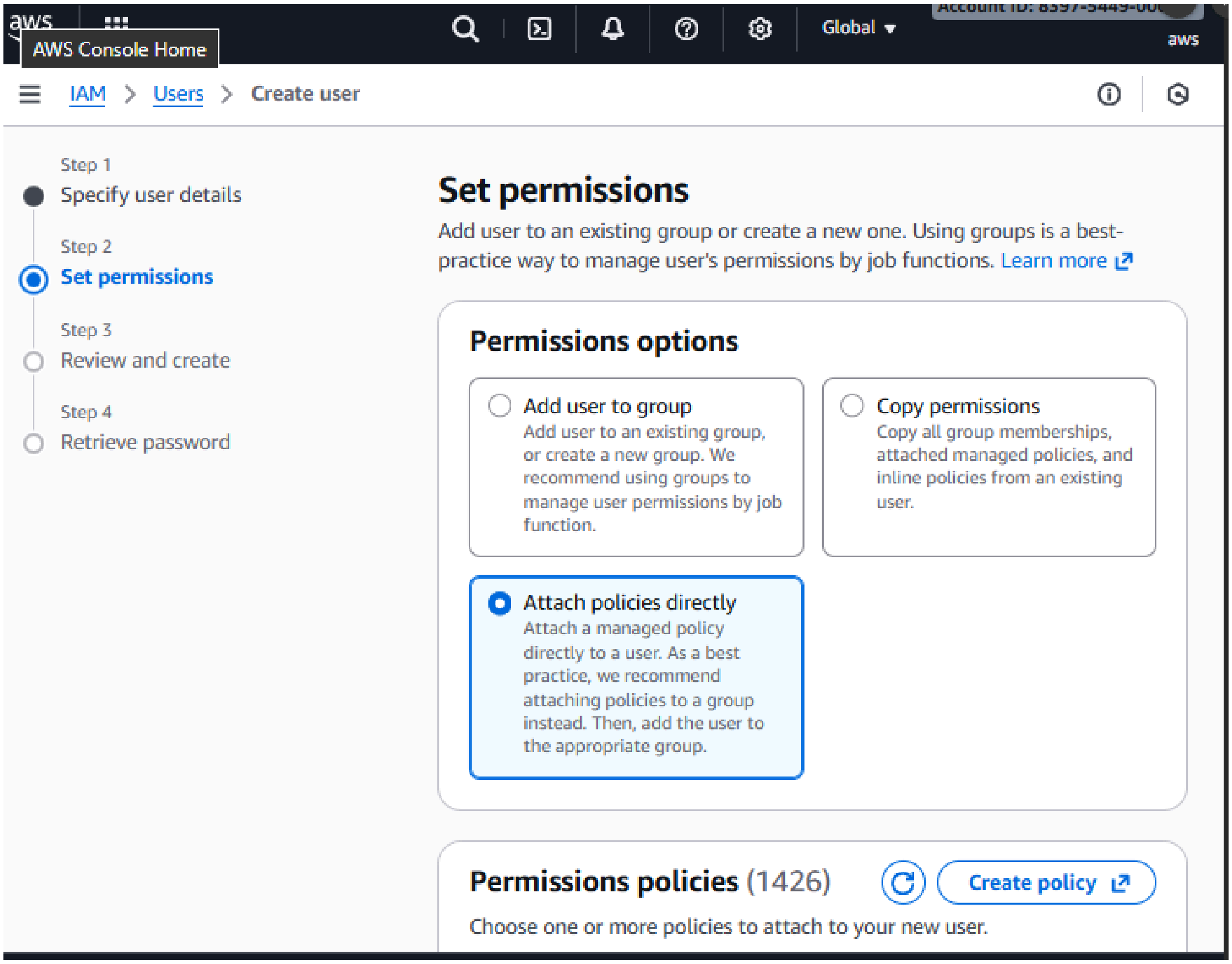
○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1426)

Choose one or more policies to attach to your new user.



Step 1
● Specify user details

Step 2
● Set permissions

Step 3
○ Review and create

Step 4
○ Retrieve password

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

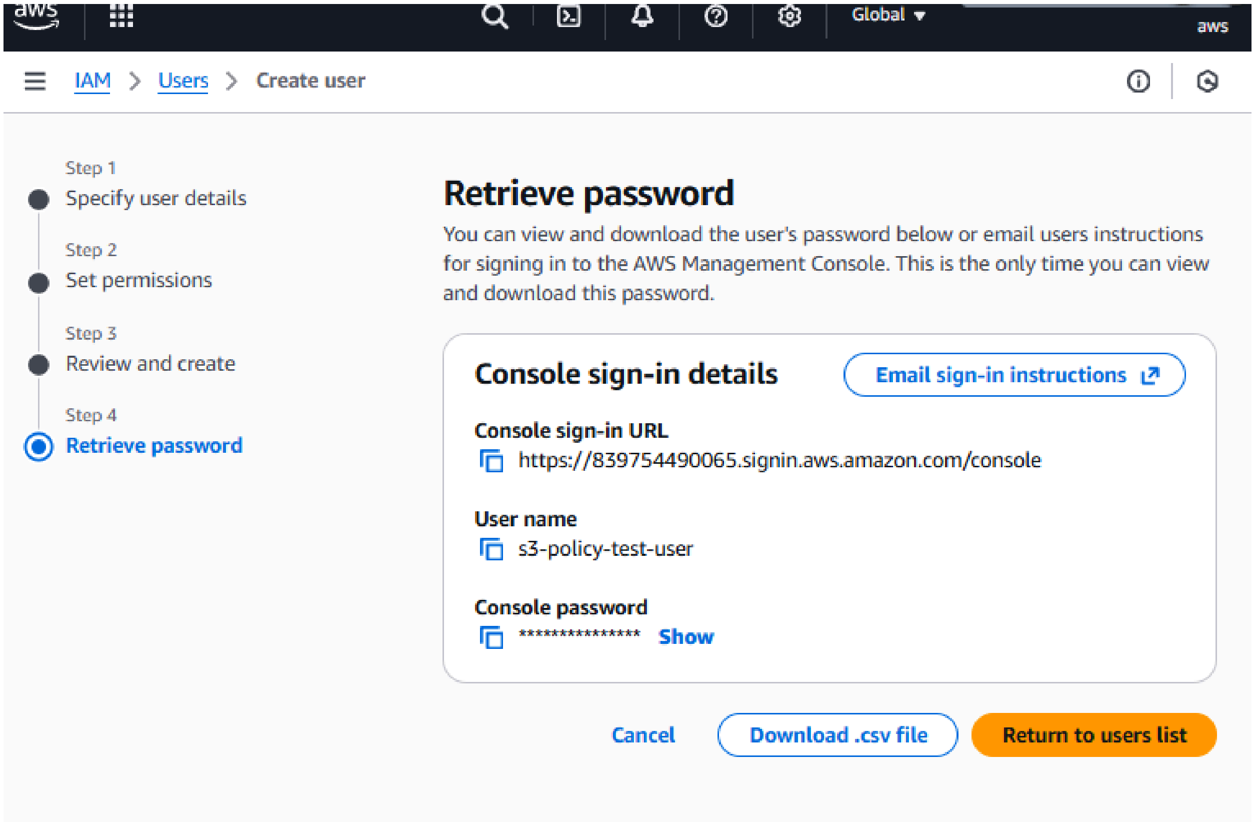
○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1426)

Choose one or more policies to attach to your new user.



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Retrieve password

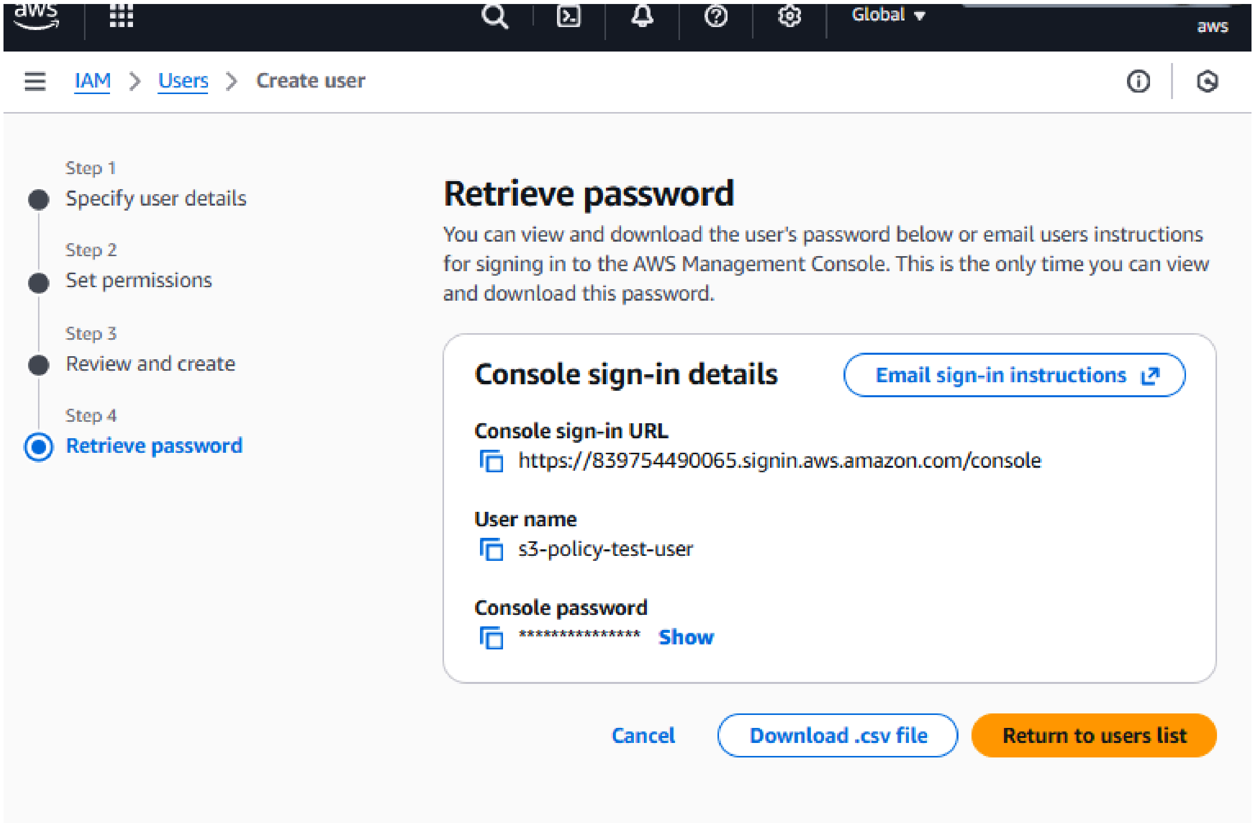
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
https://839754490065.signin.aws.amazon.com/console

User name
s3-policy-test-user

Console password
****



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Retrieve password

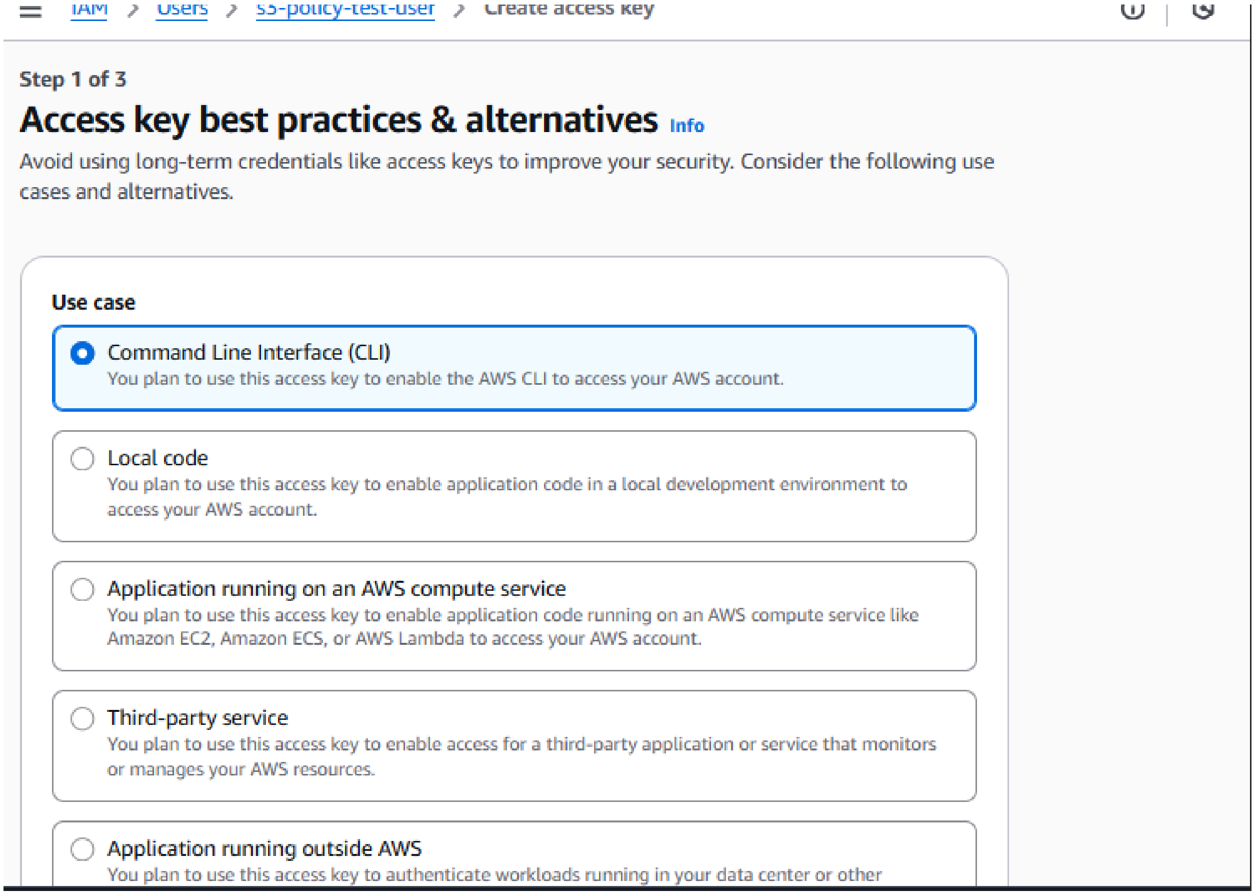
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
https://839754490065.signin.aws.amazon.com/console

User name
s3-policy-test-user

Console password
****



Step 1 of 3

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

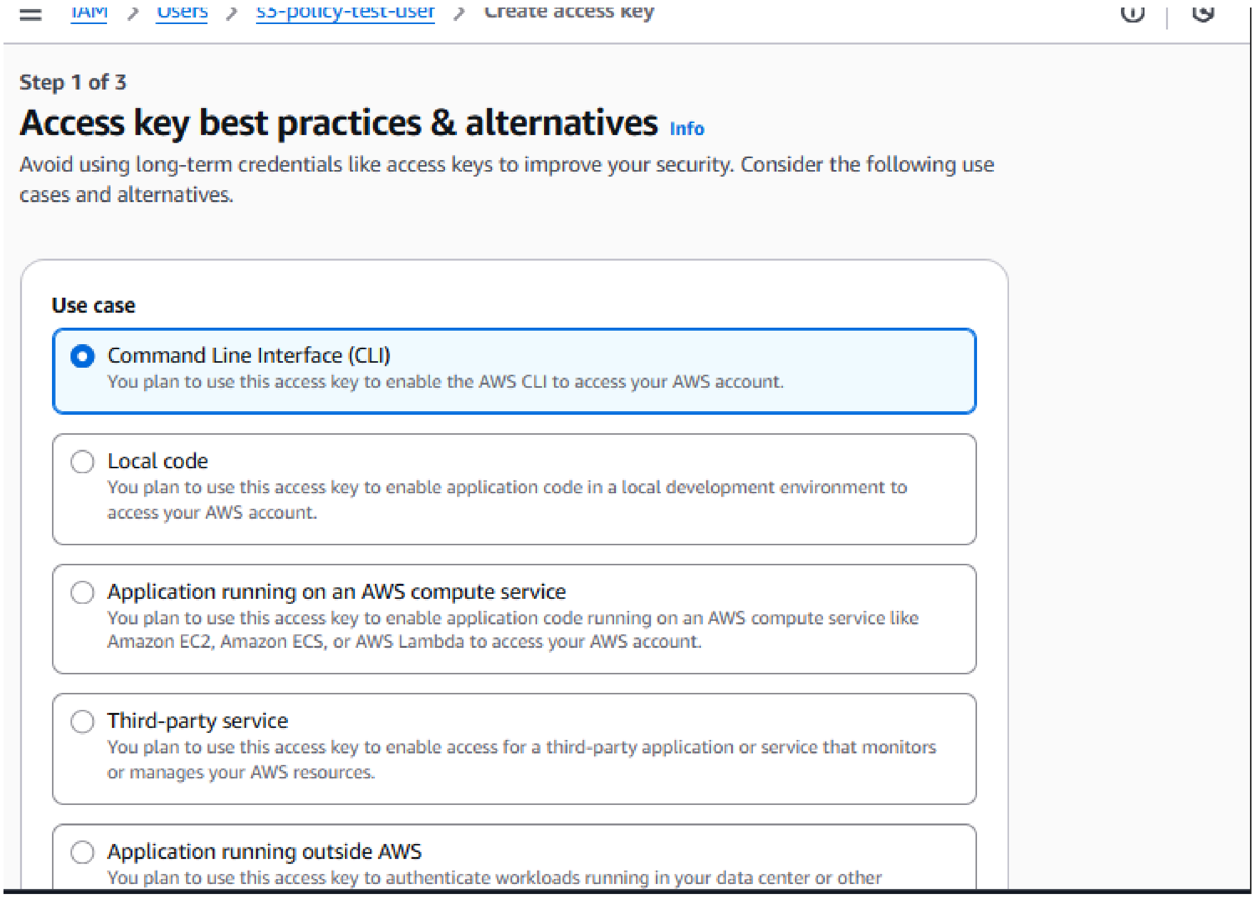
● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other



IAM > Users > s3-policy-test-user > Create access key

Step 1 of 3

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

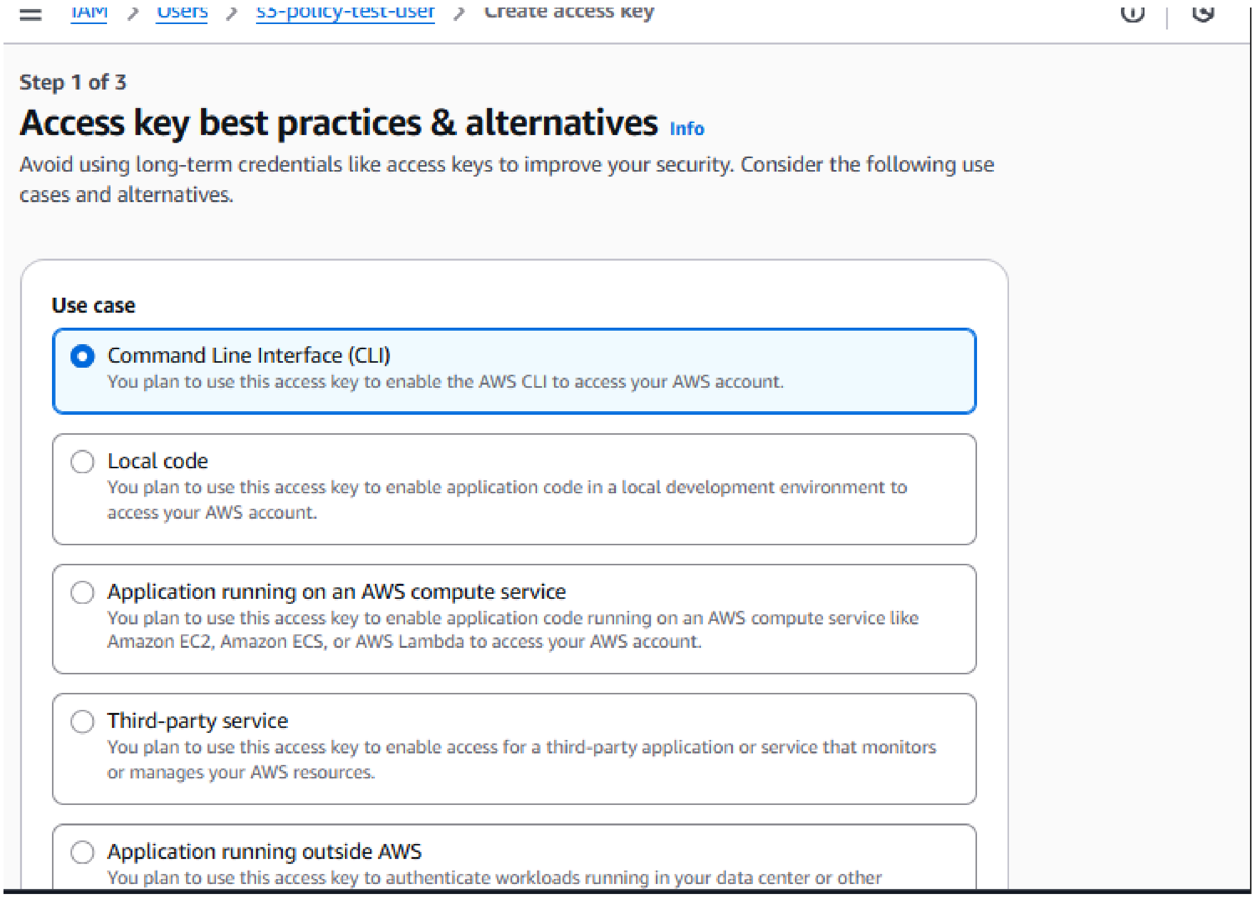
● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other



IAM > Users > s3-policy-test-user > Create access key

Step 1 of 3

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

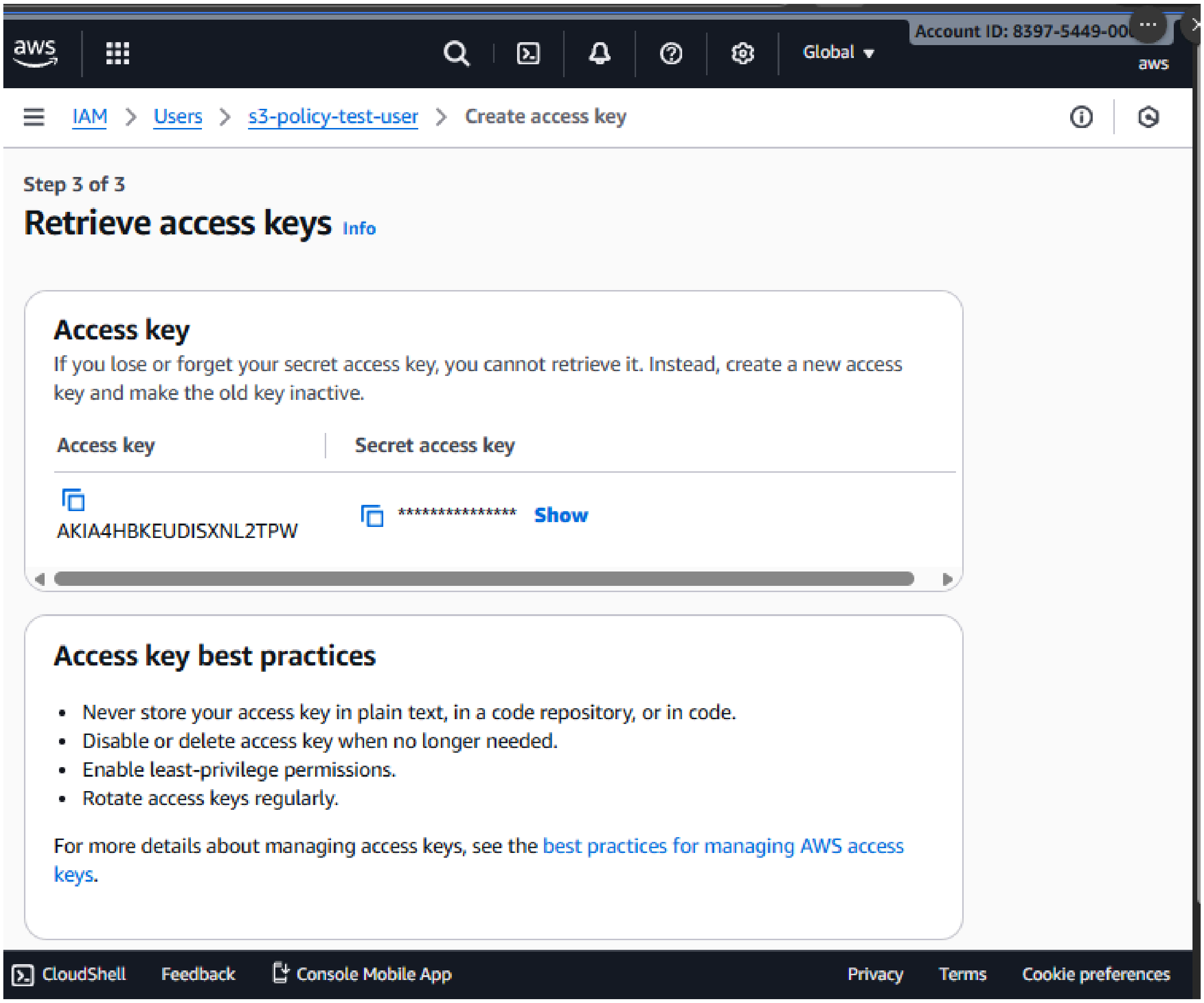
● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

○ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

○ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other



Step 3 of 3

Retrieve access keys Info

Access key

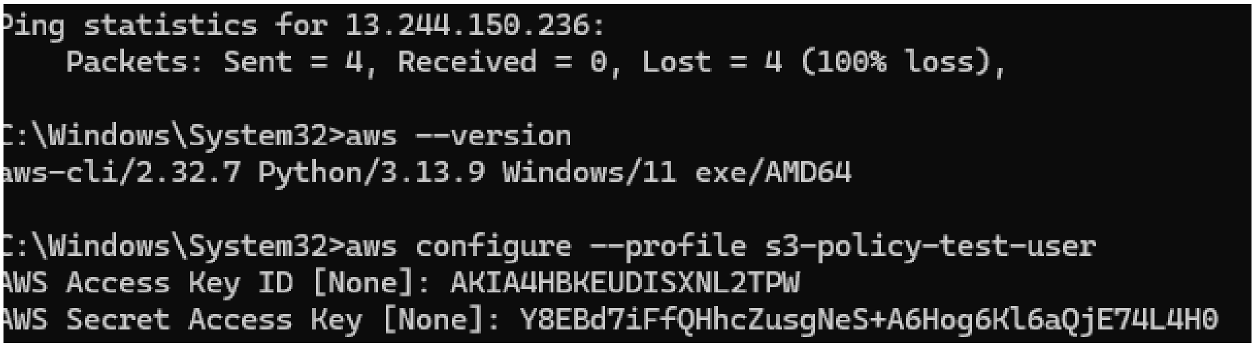
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA4HBKEUDISXNL2TPW	*************** Show

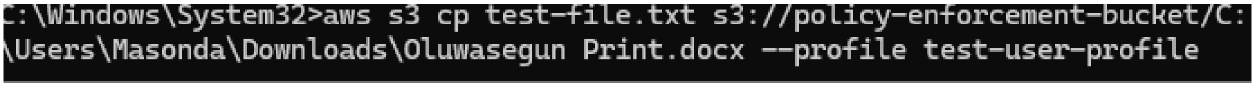
Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

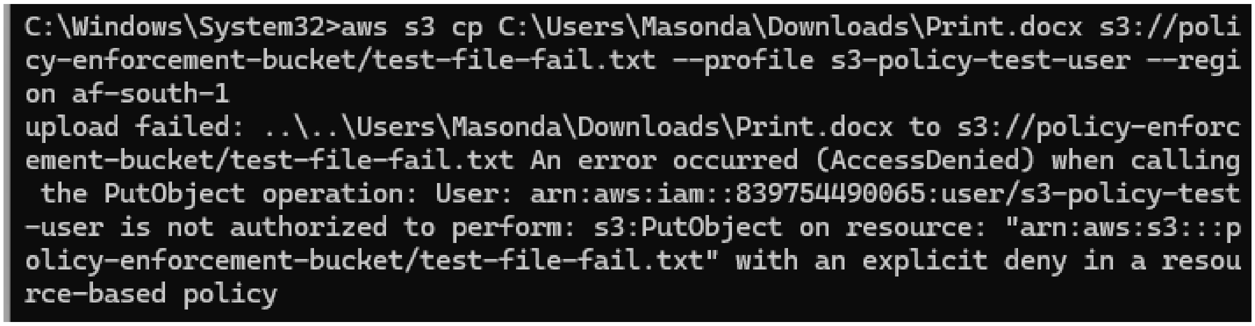
For more details about managing access keys, see the best practices for managing AWS access keys.



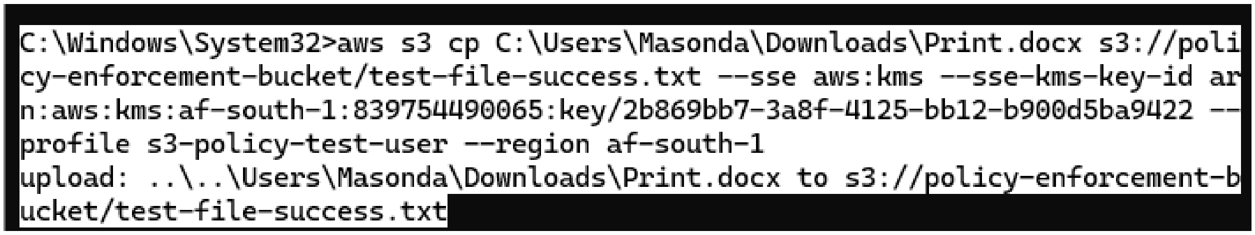
```
Ping statistics for 13.244.150.236:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Windows\System32>aws --version
aws-cli/2.32.7 Python/3.13.9 Windows/11 exe/AMD64

C:\Windows\System32>aws configure --profile s3-policy-test-user
AWS Access Key ID [None]: AKIA4HBKEUDISXNL2TPW
AWS Secret Access Key [None]: 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
```



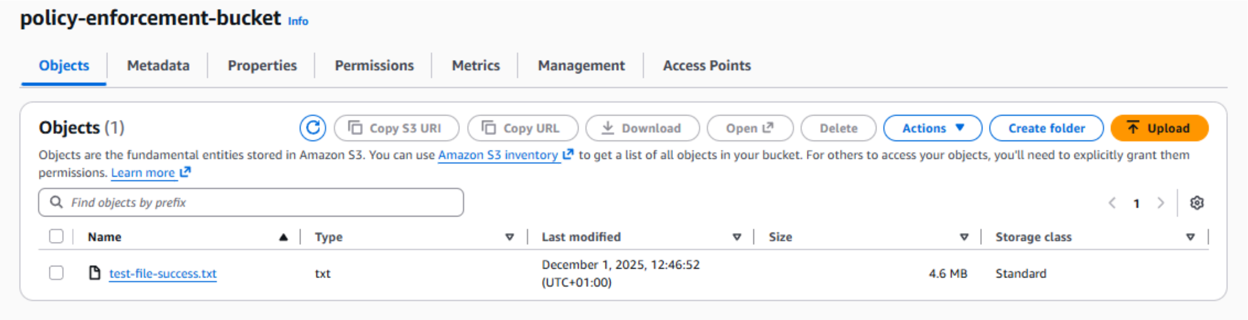
```
C:\Windows\System32>aws s3 cp test-file.txt s3://policy-enforcement-bucket/C:
\Users\Masonda\Downloads\Oluwasegun Print.docx --profile test-user-profile
```



```
C:\Windows\System32>aws s3 cp C:\Users\Masonda\Downloads\Print.docx s3://poli
cy-enforcement-bucket/test-file-fail.txt --profile s3-policy-test-user --regi
on af-south-1
upload failed: ..\..\Users\Masonda\Downloads\Print.docx to s3://policy-enforc
ement-bucket/test-file-fail.txt An error occurred (AccessDenied) when calling
 the PutObject operation: User: arn:aws:iam::839754490065:user/s3-policy-test
-user is not authorized to perform: s3:PutObject on resource: "arn:aws:s3:::p
olicy-enforcement-bucket/test-file-fail.txt" with an explicit deny in a resou
rce-based policy
```



```
C:\Windows\System32>aws s3 cp C:\Users\Masonda\Downloads\Print.docx s3://poli
cy-enforcement-bucket/test-file-success.txt --sse aws:kms --sse-kms-key-id ar
n:aws:kms:af-south-1:839754490065:key/2b869bb7-3a8f-4125-bb12-b900d5ba9422 --
profile s3-policy-test-user --region af-south-1
upload: ..\..\Users\Masonda\Downloads\Print.docx to s3://policy-enforcement-b
ucket/test-file-success.txt
```

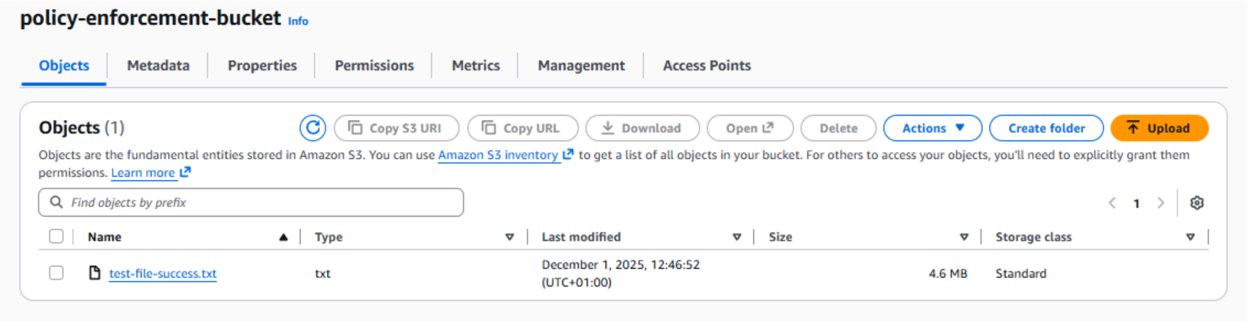


policy-enforcement-bucket Info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
test-file-success.txt	txt	December 1, 2025, 12:46:52 (UTC+01:00)	4.6 MB	Standard

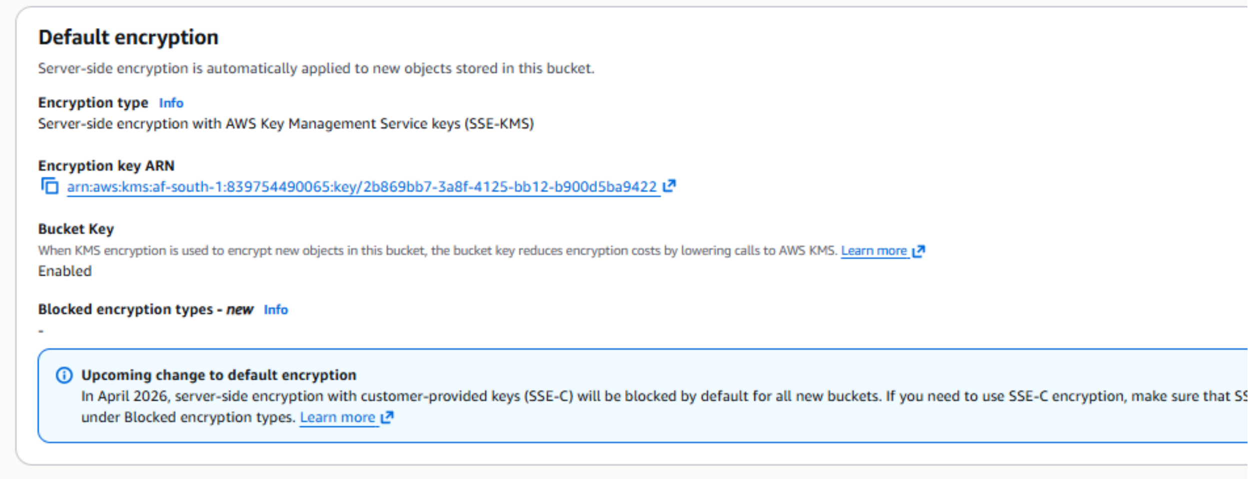


policy-enforcement-bucket Info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
test-file-success.txt	txt	December 1, 2025, 12:46:52 (UTC+01:00)	4.6 MB	Standard



Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

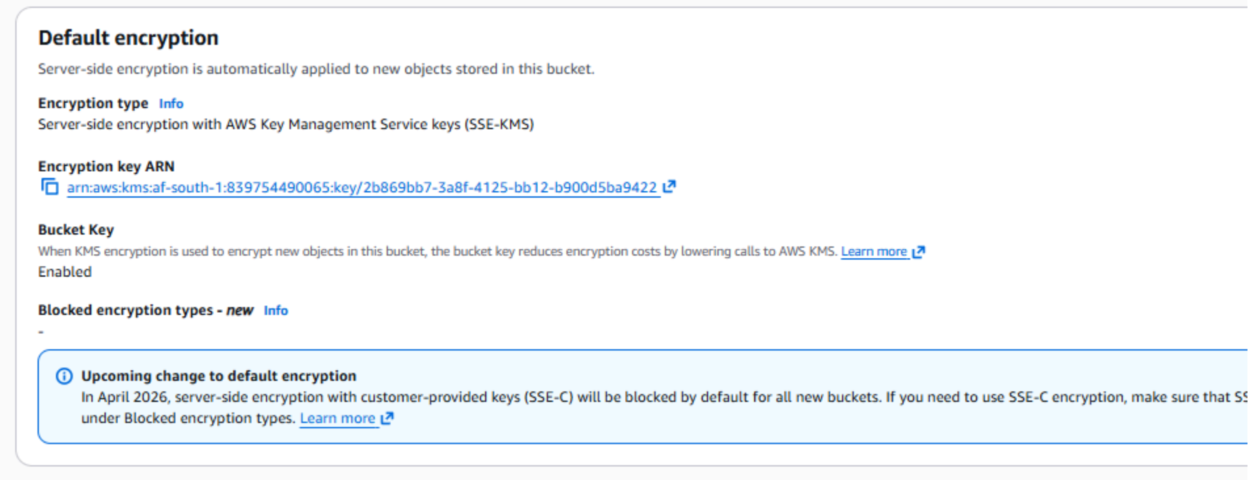
Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:af-south-1:839754490065:key/2b869bb7-3a8f-4125-bb12-b900d5ba9422

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Blocked encryption types - *new* Info
-

Upcoming change to default encryption
In April 2026, server-side encryption with customer-provided keys (SSE-C) will be blocked by default for all new buckets. If you need to use SSE-C encryption, make sure that SS
under Blocked encryption types. Learn more



Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

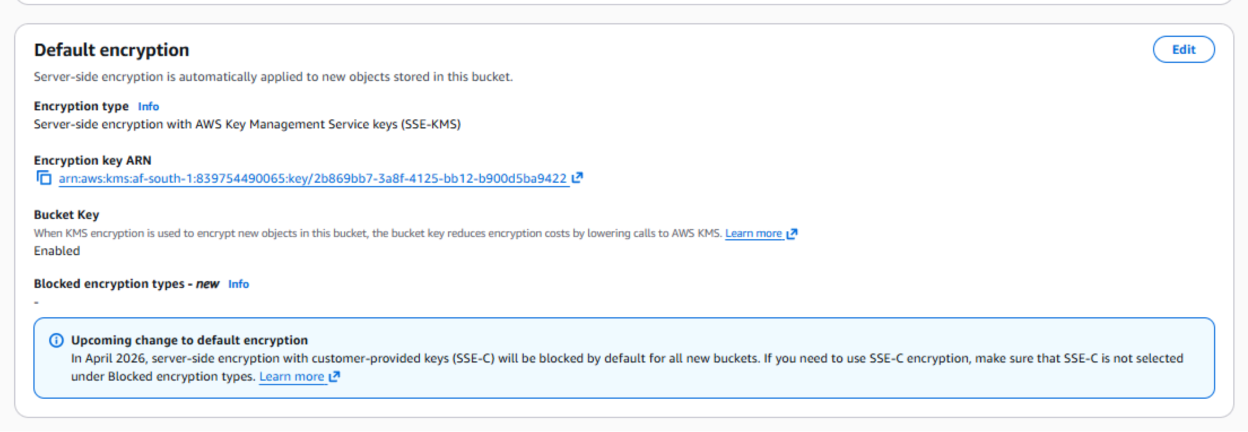
Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:af-south-1:839754490065:key/2b869bb7-3a8f-4125-bb12-b900d5ba9422

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Blocked encryption types - *new* Info
-

Upcoming change to default encryption
In April 2026, server-side encryption with customer-provided keys (SSE-C) will be blocked by default for all new buckets. If you need to use SSE-C encryption, make sure that SS
under Blocked encryption types. Learn more



Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

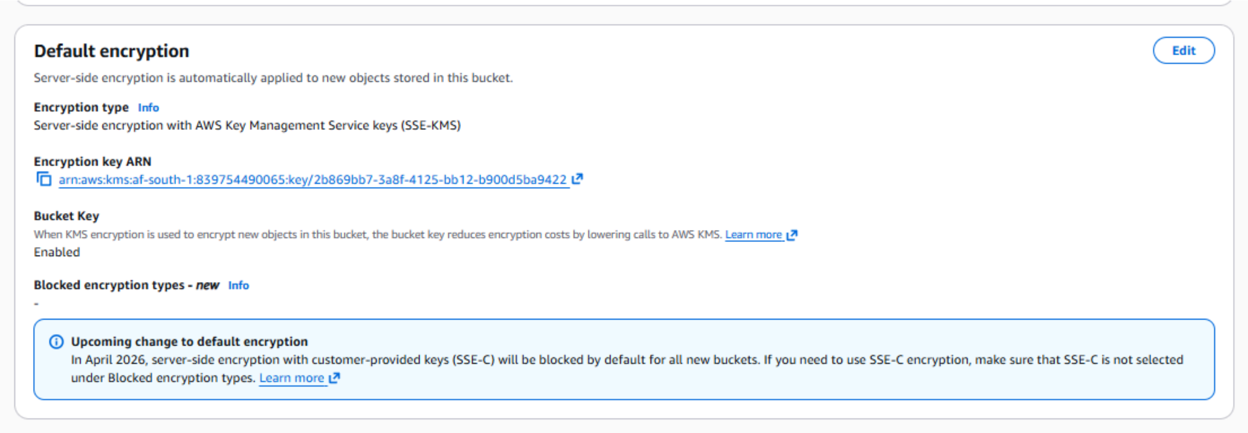
Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:af-south-1:839754490065:key/2b869bb7-3a8f-4125-bb12-b900d5ba9422

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Blocked encryption types - *new* Info
-

Upcoming change to default encryption
In April 2026, server-side encryption with customer-provided keys (SSE-C) will be blocked by default for all new buckets. If you need to use SSE-C encryption, make sure that SSE-C is not selected under Blocked encryption types. Learn more



Default encryption

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Encryption key ARN
arn:aws:kms:af-south-1:839754490065:key/2b869bb7-3a8f-4125-bb12-b900d5ba9422

Bucket Key
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more
Enabled

Blocked encryption types - *new* Info
-

Upcoming change to default encryption
In April 2026, server-side encryption with customer-provided keys (SSE-C) will be blocked by default for all new buckets. If you need to use SSE-C encryption, make sure that SSE-C is not selected under Blocked encryption types. Learn more