

services

s3

Create bucket

- 2 ways to enable encryption:
1. Via console
  2. S3 bucket policy

Create bucket

1 Name and region 2 Set properties 3 Set permissions 4 Review

Name and region

Bucket name [?](#)

Region

Copy settings from an existing bucket

0 Buckets

Create Cancel Next

Create bucket

1 Name and region 2 Configure options 3 Set permissions 4 Review

Tags

You can use tags to track project costs. [Learn more](#)

[+ Add another](#)

Object-level logging

☐ Record object-level API activity using AWS CloudTrail for an additional cost. See [CloudTrail pricing](#) or [learn more](#)

Default encryption

☒ Automatically encrypt objects when they are stored in S3. [Learn more](#)

☒ AES-256  
Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

☐ AWS-KMS  
Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Management

CloudWatch request metrics

☐ Monitor requests in your bucket for an additional cost. See [CloudWatch pricing](#) or [learn more](#)

Previous Next

# Encryption using s3 bucket policy

1 services

2 s3

3 Create bucket by checking default encryption

Search for buckets

+ Create bucket Delete bucket Empty bucket

4

Bucket name	Access	Region	Date created
fayes-encrypted-files	Not public *	EU (Frankfurt)	Apr 25, 2018 12:46:20 PM GMT+0100

\* Objects might still be publicly accessible due to object ACLs. [Learn more](#)

Amazon S3 > fayes-encrypted-files

Overview Permissions Management

Bucket policy use JSON-based access policy language to manage advanced permission to your Amazon S3 resources.

Access Control List Bucket Policy CORS configuration

Access for your AWS account

Account	List objects	Write objects	Read bucket
<input type="radio"/> d18d1b66545ce16bc6b50a1d593e3059b978b32cd12d9683f39bec66cb185729	Yes	Yes	Yes

Overview Properties Permissions Management

Access Control List Bucket Policy CORS configuration

6

Bucket policy editor ARN: arn:aws:s3:::fayes-encrypted-files

Type to add a new policy or edit an existing policy in the text area below.

1 2

Documentation Policy generator

# AWS Policy Generator 7

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

## Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

## Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect ☐ Allow ☒ Deny

Principal

AWS Service Amazon S3 ☐ All Services ("\*")

Actions 1 Action(s) Selected ☐ All Actions ("\*")

Amazon Resource Name (ARN) arn:aws:s3:::fayes-encrypted-files

ARN should follow the following format: arn:aws:s3:::<bucket\_name>/<key\_name>.

Add Conditions (Optional)

Add Statement

PutObject

\* → User who uses this bucket to upload files without server-side encryption is going to be denied

"Resource": "arn:aws:s3:::fayes-encrypted-files/\*",

some services do not let you specify specific actions for individual resources.  
/\* action will apply to all resources within that service and not just the bucket

# 8

## Add Conditions (Optional)

Conditions are any restrictions or details about the statement. [\(More Details\)](#).

Condition StringNotEquals

Key s3:x-amz-server-side-encryption

Value aws:kms

Add Condition

Add Statement

# 9

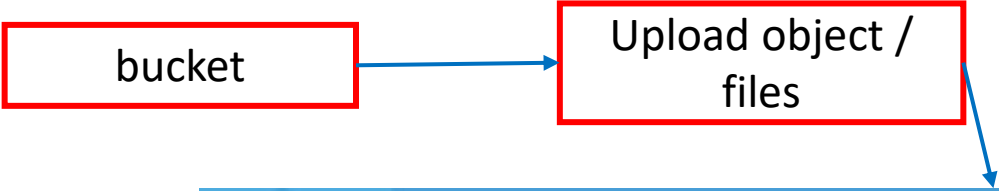
Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will not be reflected in the policy generator tool.

```
{
  "Id": "Policy1524656917361",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1524656913941",
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::fayes-encrypted-files",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "aws:kms"
        }
      },
      "Principal": "*"
    }
  ]
}
```

Close

Testing the bucket policy added



The screenshot shows the AWS S3 Upload console. At the top, there's a blue header with the word "Upload" and a close button. Below the header, there are four steps: "1 Select files", "2 Set permissions", "3 Set properties", and "4 Review". Step 3 is highlighted with a red box. Below the steps, there are two radio button options for storage class: "One Zone-IA" (selected) and "Reduced Redundancy (Not recommended)". Below these, there's an "Encryption" section with three radio button options: "None" (selected), "Amazon S3 master-key", and "AWS KMS master-key". The "None" option is highlighted with a red box. Below the encryption section, there's a "Metadata" section with a table for headers and tags. At the bottom, there are "Upload", "Previous", and "Next" buttons.

Header	Value
Select a key	

Key	Value
Key	Value

None – Forbidden error  
Amazon s3 master-key – forbidden error  
AWS KMS master key – file gets uploaded to s3

It works only for the specified encryption. When we try to upload to the S3 bucket without specified encryption – we get forbidden error.