

1. Create an S3 bucket and upload some objects to S3.

Login to aws a/c and goto s3 service and click on Create bucket:

The screenshot shows the AWS S3 Buckets page. At the top, there are tabs for 'General purpose buckets' (which is selected) and 'Directory buckets'. Below the tabs, it says 'General purpose buckets (3)' with a 'Info' link. To the right of this are buttons for 'Copy ARN', 'Empty', 'Delete', and a prominent orange 'Create bucket' button. The URL in the browser bar is 'Amazon S3 > Buckets'.

Give the bucket name which should be unique across the aws accounts:

The screenshot shows the 'Create bucket' page. In the 'General configuration' section, the 'Bucket name' field contains 's-horizon-bucket2'. Under 'Object Ownership', the 'ACLs disabled (recommended)' option is selected. The URL in the browser bar is 'Amazon S3 > Buckets > Create bucket'.

Block the public access and disable the versioning:

The screenshot shows the 'Create bucket' page with 'Block Public Access settings for this bucket' and 'Bucket Versioning' sections. In the 'Block Public Access' section, 'Block all public access' is checked. In the 'Bucket Versioning' section, 'Disable' is selected. The URL in the browser bar is 'Amazon S3 > Buckets > Create bucket'.

And create the bucket:

Default encryption [Info](#)
 Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
 Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Bucket Key
 Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

Goto the bucket and click on upload:

Amazon S3 > Buckets > s-horizon-bucket2

s-horizon-bucket2 [Info](#)

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (0)
 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](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

[Upload](#)

Click on Add file/Add folder:

Amazon S3 > Buckets > s-horizon-bucket2 > Upload

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose [Add files](#) or [Add folder](#).

Files and folders (0)
 All files and folders in this table will be uploaded.

Name	Folder	Type	Size
No files or folders You have not chosen any files or folders to upload.			

Destination [Info](#)
Destination [s3://s-horizon-bucket2](#)

Destination details
 Bucket settings that impact new objects stored in the specified destination.

Permissions
 Grant public access and access to other AWS accounts.

Properties
 Specify storage class, encryption settings, tags, and more.

[Cancel](#) [Upload](#)

Add any file/folder and upload:

The screenshot shows the AWS S3 'Upload' interface. At the top, the path is 'Amazon S3 > Buckets > s-horizon-bucket2 > Upload'. The main area is titled 'Upload Info' with a note: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more.' A large dashed box allows dragging and dropping files. Below it, a table lists 'Files and folders (1 total, 54.0 B)'. The table has columns: Name, Folder, Type, Size, Status, and Error. One item, 'test.txt', is listed with type 'text/plain' and size '54.0 B'. Buttons for 'Remove', 'Add files', and 'Add folder' are at the top right. A 'Destination' section shows 's3://s-horizon-bucket2'. A 'Destination details' link is provided. At the bottom right are 'Cancel' and 'Upload' buttons, with 'Upload' highlighted.

Check the file got uploaded:

The screenshot shows the 'Upload: status' page. A green banner at the top says 'Upload succeeded' with a note: 'For more information, see the Files and folders table.' A blue info icon says 'After you navigate away from this page, the following information is no longer available.' Below is a 'Summary' table with two rows: 'Destination s3://s-horizon-bucket2' under 'Succeeded' (1 file, 54.0 B (100.00%)) and 'Failed' (0 files, 0 B (0%)). A 'Close' button is in the top right. At the bottom, tabs for 'Files and folders' (selected) and 'Configuration' are shown. The 'Files and folders' table is identical to the one in the upload interface, listing 'test.txt' with status 'Succeeded'.

2. Deploy a static website in the S3 bucket.

Goto the bucket->Properties and click on 'Static Website Hosting' ->Edit:

Amazon S3 > Buckets > s-horizon-bucket2

Transfer acceleration
Use an accelerated endpoint for faster data transfers. [Learn more](#) [Edit](#)

Transfer acceleration
Disabled

Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#) [Edit](#)

Object Lock
Disabled

Requester pays
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#) [Edit](#)

Requester pays
Disabled

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#) [Edit](#)

We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about [Amplify Hosting](#) or [View your existing Amplify apps](#) [Create Amplify app](#)

S3 static website hosting
Disabled

Enable the 'static website hosting' and give the 'index.html' and 'error.html' and save:

Amazon S3 > Buckets > s-horizon-bucket2 > Edit static website hosting

Edit static website hosting [Info](#)

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Disable
 Enable

Hosting type
 Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document
Specify the home or default page of the website.
index.html

Error document - optional
This is returned when an error occurs.
error.html

Redirection rules - optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

1	
---	--

Create a file with name 'index.html' and upload to the above created bucket:

Amazon S3 > Buckets > s-horizon-bucket2

s-horizon-bucket2 [Info](#)

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (2)
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](#)

<input type="checkbox"/> Name	Type	Last modified	Size	Storage class
 index.html	html	September 29, 2025, 15:53:28 (UTC+05:30)	3.1 KB	Standard
 test.txt	txt	September 29, 2025, 15:40:19 (UTC+05:30)	54.0 B	Standard

[Actions](#) [Create folder](#) [Upload](#)

Then remove the checkbox from ‘Block public access’, :

And remove the ACL's and attach policy:

Goto the properties of the bucket and in the down side we get the url:

Will get the static website:

-----done-----

3. Enable cross-region replication on S3 buckets.

Create 2 buckets in 2 different Regions and enable the Versioning for both buckets.

Create IAM role with 'aws service' and give the service as 's3' and click on Next:

The screenshot shows the 'Select trusted entity' step of the IAM role creation process. It includes a navigation bar at the top with 'IAM > Roles > Create role'. On the left, a sidebar lists 'Step 1 Select trusted entity' (selected), 'Step 2 Add permissions', and 'Step 3 Name, review, and create'. The main area is titled 'Select trusted entity' with a 'Trusted entity type' section containing five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is a 'Use case' section with a dropdown set to 'S3' and a list of two options: 'S3' (selected) and 'S3 Batch Operations'. At the bottom right are 'Cancel' and 'Next' buttons.

and select the policy 'AmazonS3FullAccess':

The screenshot shows the 'Add permissions' step of the IAM role creation process. It includes a navigation bar at the top with 'IAM > Roles > Create role'. On the left, a sidebar lists 'Step 1 Select trusted entity' (selected), 'Step 2 Add permissions' (selected), and 'Step 3 Name, review, and create'. The main area is titled 'Add permissions' with a 'Permissions policies (1/1075)' section showing one policy selected: 'AmazonS3FullAccess'. A 'Filter by Type' search bar is present. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

Give the role name:

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
`Cross_Region_Replication`

Description
Add a short explanation for this role.
Allows S3 to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

```

1+ [ {
2+   "Version": "2012-10-17",
3+   "Statement": [
4+     {
5+       "Effect": "Allow",
6+       "Principal": "*",
7+       "Service": "s3.amazonaws.com"
8+     },
9+     {
10+      "Action": "sts:AssumeRole"
11+    }
12+  ]
}

```

And click on 'Create Role':

Step 1: Select trusted entities

Trust policy

```

1+ [ {
2+   "Version": "2012-10-17",
3+   "Statement": [
4+     {
5+       "Effect": "Allow",
6+       "Principal": "*",
7+       "Service": "s3.amazonaws.com"
8+     },
9+     {
10+      "Action": "sts:AssumeRole"
11+    }
12+  ]
}

```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
<code>AmazonS3FullAccess</code>	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Previous](#) **Create role**

And Create another policy and attach to this role :

```

"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetReplicationConfiguration",
            "s3>ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
        ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    }
]
}

```

Note: here replace the name of the bucket of source and destination with your own source and destination buckets, below is also same policy of json:

```
{

```

```
"Version":"2012-10-17",

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetReplicationConfiguration",
      "s3>ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-source-bucket"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3GetObjectVersionForReplication",
      "s3GetObjectVersionAcl",
      "s3GetObjectVersionTagging"
    ],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
```

```

    "s3:ReplicateObject",
    "s3:ReplicateDelete",
    "s3:ReplicateTags"
],
"Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
]
}

```

Activating Cross region Replication

- Click on the Source bucket. Go to the management and Click on the management tab.
- Create Replication Rule, give rule name and source bucket choose ‘Apply to all objects in the bucket’ because I want to apply the rule to all the objects in the buckets:

Amazon S3 > Buckets > s-horizon-bucket2 > Replication rules > Create replication rule

Create replication rule Info

Replication rule configuration

Replication rule name
SourceRepRule1

Status
 Enabled
 Disabled

Priority
The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.
0

Source bucket

Source bucket name
s-horizon-bucket2

Source Region
US East (N. Virginia) us-east-1

Choose a rule scope
 Limit the scope of this rule using one or more filters
 Apply to all objects in the bucket

Scroll down and choose the Destination :choose first option since the destination bucket is also in the same account:

Destination

Destination
You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#) or see [Amazon S3 pricing](#).

Choose a bucket in this account
 Specify a bucket in another account

Click on Browse S3 and choose your destination bucket:

The screenshot shows two sequential steps in the AWS S3 console.

Step 1: Choose a bucket

This dialog lists available S3 buckets across different AWS regions. A specific bucket, "s-horizon-bucket-ohio", is selected and highlighted with a blue border. The bucket name is "s-horizon-bucket-ohio", located in "US East (Ohio) us-east-2". Other buckets listed include "s-abhi-bucket-1", "s-horizon-bucket1", "s-horizon-bucket2", and "shuja-techi-bucket123".

Step 2: Destination

This dialog is used to configure replication rules. It includes sections for "Destination", "Bucket name", and "Destination Region". Under "Destination", the "Choose a bucket in this account" option is selected. Under "Bucket name", the value "s-horizon-bucket-ohio" is entered. Under "Destination Region", "US East (Ohio) us-east-2" is chosen.

Next, select the IAM role: [the role which has created above :

This screenshot shows the "IAM role" selection step. It displays a message stating "The selected IAM role applies to all rules in this configuration." Below this, the "Cross_Region_Replication" role is listed under "IAM role".

Upload files to Source Bucket to Replicate to the Destination Bucket.

This screenshot shows the "Objects" list for the "s-horizon-bucket2" bucket. The list contains three objects: "abc.png" and "out.png". Both files were uploaded on September 29, 2025, at 15:40:19 (UTC+05:30). The file "abc.png" is 54.0 KB in size and has a Standard storage class. The file "out.png" is 109.0 KB in size and also has a Standard storage class.

- If you now check in the two buckets, files are uploaded successfully in the source bucket as well as the destination bucket due to CRR.

The screenshot shows the AWS S3 console interface. The top navigation bar includes links for EC2, API Gateway, IAM, Route 53, VPC, and S3. The main navigation bar shows 'Amazon S3 > Buckets > s-horizon-bucket-ohio'. The left sidebar under 'General purpose buckets' lists various options like Directory buckets, Table buckets, Vector buckets, Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, and Batch Operations. The main content area is titled 's-horizon-bucket-ohio Info' and shows the 'Objects' tab selected. It displays one object, 'out.png', which is a PNG file uploaded on September 29, 2025, at 18:25:09 UTC+05:30, with a size of 109.0 KB and a storage class of Standard. There are buttons for Actions (Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, Upload), a search bar for 'Find objects by prefix', and a table header for Name, Type, Last modified, Size, and Storage class.

-----done-----

4. Configure a bucket policy so only the Admin user can see the objects of the S3 bucket.

Create a bucket/to existing bucket add the below policy :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAdminFullRead",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "arn:aws:iam::931208603575:user/testuser"  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3>ListBucket",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::s-horizonbucket-3",  
                "arn:aws:s3:::s-horizonbucket-3/*"  
            ]  
        }  
    ]  
}
```

```
{
    "Sid": "DenyAllExceptAdmin",
    "Effect": "Deny",
    "NotPrincipal": {
        "AWS": "arn:aws:iam::931208603575:user/testuser"
    },
    "Action": [
        "s3:GetObject",
        "s3>ListBucket",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:s3:::s-horizonbucket-3",
        "arn:aws:s3:::s-horizonbucket-3/*"
    ]
}
```

With user 'testuser' is able to access all the buckets.

The screenshot shows the AWS S3 console with the path **Buckets > s-horizonbucket-3**. On the left sidebar, under **General purpose buckets**, there is a list of various AWS services and features. The main panel displays the **s-horizonbucket-3** bucket's details. The **Objects** tab is selected, showing a single object named **put.png** with a size of 109.0 KB and a storage class of Standard. The last modified date is September 30, 2025, at 12:42:34 UTC+05:30.

Login with 'testuser2' which cannot access the objects of the bucket:

The screenshot shows the AWS S3 console with the path **Buckets > s-horizonbucket-3**. The user is identified as **testuser2** in the top right corner. The interface is identical to the previous screenshot, but the **Objects** tab shows an error message: **Insufficient permissions to list objects**. The message explains that after permissions are updated, the user needs to refresh the page. A **Diagnose with Amazon Q** button is also present.

-----done-----

5. Set up lifecycle policies to automatically transition or delete objects based on specific criteria.

Select your bucket and Goto Management tab:

Lifecycle configuration

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Lifecycle rules

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. [Learn more](#)

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete mar...	Incomplete multipart up...

No lifecycle rules

There are no lifecycle rules for this bucket.

[Create lifecycle rule](#)

Give the name for the rule and select any filter like if you want for specific objects or all objects you want to do transition, here I want to do for all objects object:

Learn more' and a checkbox 'I acknowledge that this rule will apply to all objects in the bucket' is checked."/>

Create lifecycle rule [Info](#)

Lifecycle rule configuration

Lifecycle rule name

tier-ia-glacier-delete

Up to 255 characters

Choose a rule scope

Limit the scope of this rule using one or more filters

Apply to all objects in the bucket

⚠️ Apply to all objects in the bucket

If you want the rule to apply to specific objects, you must use a filter to identify those objects. Choose "Limit the scope of this rule using one or more filters". [Learn more](#)

I acknowledge that this rule will apply to all objects in the bucket.

And select which version you want to do transition: here I want to do transition for current version:

Lifecycle rule actions

Choose the actions you want this rule to perform.

Transition current versions of objects between storage classes

This action will move current versions.

Transition noncurrent versions of objects between storage classes

This action will move noncurrent versions.

Expire current versions of objects

Permanently delete noncurrent versions of objects

Delete expired object delete markers or incomplete multipart uploads

These actions are not supported when filtering by object tags or object size.

⚠️ Transitions are charged per request

For a lifecycle transition action, each request corresponds to an object transition. For details on lifecycle transition pricing, see requests pricing info on the [Storage & requests tab of the Amazon S3 pricing page](#).

I acknowledge that this lifecycle rule will incur a transition cost per request.

ⓘ By default, objects less than 128KB will not transition across any storage class

We don't recommend transitioning objects less than 128 KB because the transition costs can outweigh the storage savings. If your use case requires transitioning objects less than 128 KB, specify a minimum object size filter for each applicable lifecycle rule with a transition action.

Select your required storage class and no. of days after object creation:

Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions	Days after object creation	Action
Standard-IA	50	Remove
Glacier Deep Archive	180	Remove
Add transition		

And click on Create rule:

Review transition and expiration actions

Current version actions	Noncurrent versions actions
Day 0 <ul style="list-style-type: none"> Objects uploaded 	Day 0 No actions defined.
Day 50 <ul style="list-style-type: none"> Objects move to Standard-IA 	
Day 180 <ul style="list-style-type: none"> Objects move to Glacier Deep Archive 	

[Cancel](#) [Create rule](#)

Then a transition rule will get created which is Enabled:

The rule "tier-ia-glacier-delete" has been successfully added and the lifecycle configuration has been updated
It may take some time for the configuration to be updated. Refresh the lifecycle rules list if changes to the configuration aren't displayed.

Lifecycle configuration

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Default minimum object size for transitions
All storage classes 128K

Lifecycle rules (1)	Actions										
tier-ia-glacier-delete	<input checked="" type="radio"/>	View details	Edit	Delete	Actions	Create lifecycle rule					
Find lifecycle rules by name											
<input checked="" type="radio"/> tier-ia-glacier-delete		<input checked="" type="radio"/> Enabled	Entire bucket	Transition to Standard-IA, then -							

-----done-----

6. Push some objects to S3 using the AWS CLI.

Configure credentials / profile

Use aws configure for a default profile, or set a named profile (recommended for multiple accounts):

Using a named profile:

aws configure --profile myprofile

Or export environment variables (temporary / CI usage):

```
export AWS_ACCESS_KEY_ID="AKIA..."  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCY"  
export AWS_DEFAULT_REGION="ap-south-1"  
# on Windows PowerShell:  
# $env:AWS_ACCESS_KEY_ID="AKIA..."  
Use --profile myprofile on commands to use a named profile.
```

Create an S3 bucket

```
aws s3 mb s3://my-bucket-name --region <region> --profile myprofile  
aws s3 mb s3:::/
```

```
dell@DESKTOP-60RBKUF MINGW64 /d  
$ aws s3 mb s3://s-horizon-bucket-2 --region us-east-1  
make_bucket: s-horizon-bucket-2  
  
dell@DESKTOP-60RBKUF MINGW64 /d  
$ aws s3 ls  
2025-06-23 07:47:59 s-abhi-bucket-1  
2025-10-01 12:49:41 s-horizon-bucket-1  
2025-10-01 14:57:12 s-horizon-bucket-2  
2025-09-29 16:54:38 s-horizon-bucket-ohio
```

Upload a single file (simple)

```
aws s3 cp ./localfile.txt s3://my-bucket-name/path/in/bucket/localfile.txt --profile  
myprofile
```

```
v dell@DESKTOP-60RBKUF MINGW64 /d  
$ aws s3 cp ./test.txt s3://s-horizon-bucket-2  
upload: .\test.txt to s3://s-horizon-bucket-2/test.txt
```

Common options:

- Make public (be careful!): --acl public-read
- Server-side encryption (SSE-S3): --sse AES256
- SSE with KMS: --sse aws:kms --sse-kms-key-id arn:aws:kms:...

Example with SSE and metadata:

```
aws s3 cp ./index.html s3://my-bucket-name/site/index.html \  
--content-type "text/html" \  
--metadata author=avinaash,env=prod \  
--acl public-read
```

```
--sse AES256 \
```

```
--profile myprofile
```

Upload a directory (multiple files) — sync (recommended) or recursive cp

Sync (uploads new/changed files):

```
aws s3 sync ./local-folder s3://my-bucket-name/folder-path --profile myprofile
```

```
dell@DESKTOP-60RBKUF MINGW64 /d
$ aws s3 sync ./dir1/ s3://s-horizon-bucket-2/folder/
upload: dir1\file2.txt to s3://s-horizon-bucket-2/folder/file2.txt
upload: dir1\file1.txt to s3://s-horizon-bucket-2/folder/file1.txt
upload: dir1\file3.txt to s3://s-horizon-bucket-2/folder/file3.txt
```

Recursive copy (uploads entire tree):

```
aws s3 cp ./local-folder s3://my-bucket-name/folder-path --recursive --profile myprofile
```

Filter uploads:

```
# include only .html files
```

```
aws s3 sync ./local-folder s3://my-bucket-name/folder-path --exclude "*" --include
"*.html" --profile myprofile
```

Verify uploaded objects

List objects:

```
aws s3 ls s3://my-bucket-name/folder-path/ --recursive --profile myprofile
```

```
dell@DESKTOP-60RBKUF MINGW64 /d
$ aws s3 ls s3://s-horizon-bucket-2/folder --recursive
2025-10-01 15:10:18          0 folder/file1.txt
2025-10-01 15:10:17          0 folder/file2.txt
2025-10-01 15:10:18          0 folder/file3.txt
```

Advanced: multipart / large files

aws s3 cp and sync automatically use multipart uploads for large files. For manual control use s3api multipart commands (create-multipart-upload, upload-part, complete-multipart-upload)

-----done-----

7. Write a Bash script to create an S3 bucket.

```
dell@DESKTOP-60RBKUF MINGW64 /d
$ cat create_bucket.sh
#!/bin/bash

bucket_name=s-horizon-bucket-3
aws s3 mb s3://${bucket_name} --region us-east-1
if [ $? -eq 0 ]; then
echo "Bucket created"
else
echo "Bucket not created"
fi
```

Execute the script

```
dell@DESKTOP-60RBKUF MINGW64 /d
$ ./create_bucket.sh
make_bucket: s-horizon-bucket-3
Bucket created
```

[#!/usr/bin/env bash](#)

[# Usage: ./create_s3_bucket.sh <bucket-name> <region> \[profile\]](#)

[BUCKET_NAME=\\$1](#)

[REGION=\\$2](#)

[PROFILE=\\${3:-default}](#)

[# --- Function: Print error and exit ---](#)

[error_exit\(\) {](#)

```
echo "✖ ERROR: $1"

exit 1
}

# --- Step 1: Check input ---

if [[ -z "$BUCKET_NAME" || -z "$REGION" ]]; then
    echo "Usage: $0 <bucket-name> <region> [profile]"
    exit 1
fi

# --- Step 2: Validate bucket name ---

if [[ ! $BUCKET_NAME =~ ^[a-z0-9-]{3,63}$ ]]; then
    error_exit "Invalid bucket name. Must be 3–63 chars, lowercase, numbers, dots, or hyphens."
fi

if [[ $BUCKET_NAME =~ ^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$ ]]; then
    error_exit "Bucket name cannot look like an IP address."
fi

if [[ $BUCKET_NAME =~ ^[-\.] || $BUCKET_NAME =~ [\-\]$ ]]; then
    error_exit "Bucket name cannot start or end with '.' or '-'."
fi

# --- Step 3: Check AWS CLI installed ---

if ! command -v aws &>/dev/null; then
    error_exit "AWS CLI is not installed. Install AWS CLI v2."
fi
```

```
# --- Step 4: Check AWS credentials ---  
  
if ! aws sts get-caller-identity --profile "$PROFILE" &>/dev/null; then  
    error_exit "AWS CLI credentials not configured for profile [$PROFILE]. Run: aws configure --profile $PROFILE"  
fi  
  
  
# --- Step 5: Check if bucket already exists ---  
  
EXISTING=$(aws s3api head-bucket --bucket "$BUCKET_NAME" --profile "$PROFILE" 2>&1)  
  
  
if [[ $? -eq 0 ]]; then  
    echo "[] Bucket [$BUCKET_NAME] already exists and you own it."  
    exit 0  
  
elif echo "$EXISTING" | grep -q '403'; then  
    error_exit "Bucket [$BUCKET_NAME] already exists but is owned by another account."  
  
elif echo "$EXISTING" | grep -q '404'; then  
    echo "[] Bucket [$BUCKET_NAME] does not exist, proceeding with creation..."  
else  
    error_exit "Unexpected error while checking bucket: $EXISTING"  
fi  
  
  
# --- Step 6: Create bucket ---  
  
if [[ "$REGION" == "us-east-1" ]]; then  
    CREATE_OUT=$(aws s3api create-bucket --bucket "$BUCKET_NAME" --region "$REGION" --profile "$PROFILE" 2>&1)  
else
```

```

CREATE_OUT=$(aws s3api create-bucket --bucket "$BUCKET_NAME" --region
"$REGION" \
--create-bucket-configuration LocationConstraint="$REGION" --profile "$PROFILE"
2>&1)

fi

if [[ $? -ne 0 ]]; then
    error_exit "Failed to create bucket: $CREATE_OUT"
fi

# --- Step 7: Verify bucket creation ---

if aws s3api head-bucket --bucket "$BUCKET_NAME" --profile "$PROFILE" 2>/dev/null;
then
    echo "🎉 SUCCESS: Bucket [$BUCKET_NAME] created in region [$REGION] using profile
[$PROFILE]."
else
    error_exit "Bucket creation failed verification step."
fi

-----

```

aws s3api is the **low-level interface** of the AWS CLI for S3. Unlike aws s3 (which is higher-level and easier for uploads/downloads), aws s3api exposes the **full S3 API** with all options.

Here are some **practical examples**:

1. Create a bucket

bash

```
# For us-east-1
aws s3api create-bucket --bucket my-unique-bucket --region us-east-1

# For other regions (must specify LocationConstraint)
aws s3api create-bucket \
--bucket my-unique-bucket \
--region ap-south-1 \
--create-bucket-configuration LocationConstraint=ap-south-1
```

2. List all buckets in your account

bash

```
aws s3api list-buckets
```

3. Upload (Put) an object

bash

```
aws s3api put-object \
--bucket my-unique-bucket \
--key folder1/hello.txt \
--body ./hello.txt
```

4. Download (Get) an object

bash

```
aws s3api get-object \  
  --bucket my-unique-bucket \  
  --key folder1/hello.txt \  
  ./downloaded-hello.txt
```

5. List objects in a bucket

bash

```
aws s3api list-objects \  
  --bucket my-unique-bucket \  
  --prefix folder1/
```

6. Get object metadata (HEAD request)

bash

```
aws s3api head-object \  
  --bucket my-unique-bucket \  
  --key folder1/hello.txt
```

7. Enable versioning on a bucket

bash

```
aws s3api put-bucket-versioning \  
  --bucket my-unique-bucket \  
  --versioning-configuration Status=Enabled
```

8. Block all public access

bash

 Copy code

```
aws s3api put-bucket-public-access-block \  
  --bucket my-unique-bucket \  
  --public-access-block-configuration \  
  "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true"
```

9. Apply server-side encryption (SSE-S3)

bash

```
aws s3api put-bucket-encryption \
--bucket my-unique-bucket \
--server-side-encryption-configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

👉 `aws s3` = easier, human-friendly (`cp`, `sync`, `ls`)

👉 `aws s3api` = advanced, exact control (like encryption, versioning, ACLs, policies)

-----done-----

8. Upload a 1 GB file to S3 using the CLI.

Uploading a **1 GB file** to Amazon S3 using the AWS CLI is straightforward. The good news: you don't need to worry about splitting the file — the AWS CLI automatically uses **multipart upload** for large files (anything over **8 MB** by default).

Generate a 1 GB test file (Linux/Mac)

```
de11@DESKTOP-60RBKUF MINGW64 /d
$ dd if=/dev/zero of=bigfile.bin bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 0.500256 s, 2.1 GB/s
```

Upload the 1 GB file to S3

Option A: High-level `aws s3` (easiest)

```
aws s3 cp ./bigfile.bin s3://my-bigfile-bucket/uploads/bigfile.bin
```

```
deLL@DESKTOP-60RBKUF MINGW64 /d
$ aws s3 cp ./bigfile.bin s3://s-horizon-bucket-3
upload: .\bigfile.bin to s3://s-horizon-bucket-3/bigfile.bin
```

Option B: Low-level aws s3api (fine-grained control)

```
aws s3api put-object \
--bucket my-bigfile-bucket \
--key uploads/bigfile.bin \
--body ./bigfile.bin
```

Verify upload

```
aws s3 ls s3://my-bigfile-bucket/uploads/
```

Or check metadata:

```
aws s3api head-object \
--bucket my-bigfile-bucket \
--key uploads/bigfile.bin
```

(Optional) Speed it up

- Use **multipart concurrency**:

```
aws s3 cp ./bigfile.bin s3://my-bigfile-bucket/uploads/ --expected-size 1073741824
```

- Add **--storage-class STANDARD_IA** if you want cheaper storage.
- Use **--sse AES256** for encryption.

-----completed-----