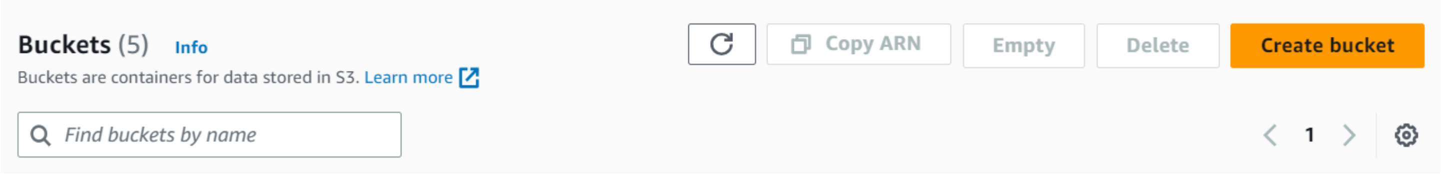
MASTERING S3 BUCKET CREATION: A COMPREHENSIVE GUIDE WITH AWS MANAGEMENT CONSOLE AND CLI

Creating an S3 Bucket using the AWS Management Console

Access the AWS Management Console: Launch your web browser and visit the AWS Management Console website at https://console.aws.amazon.com. Use your AWS account credentials to log in.

2. Access the S3 Service: After logging in, utilize the search bar within the AWS Management Console to search for "S3". Click on the "Amazon S3" service that appears in the search results.

3. Initiate Bucket Creation: Within the S3 console, locate and click on the "Create bucket" button to commence the creation of a new bucket.

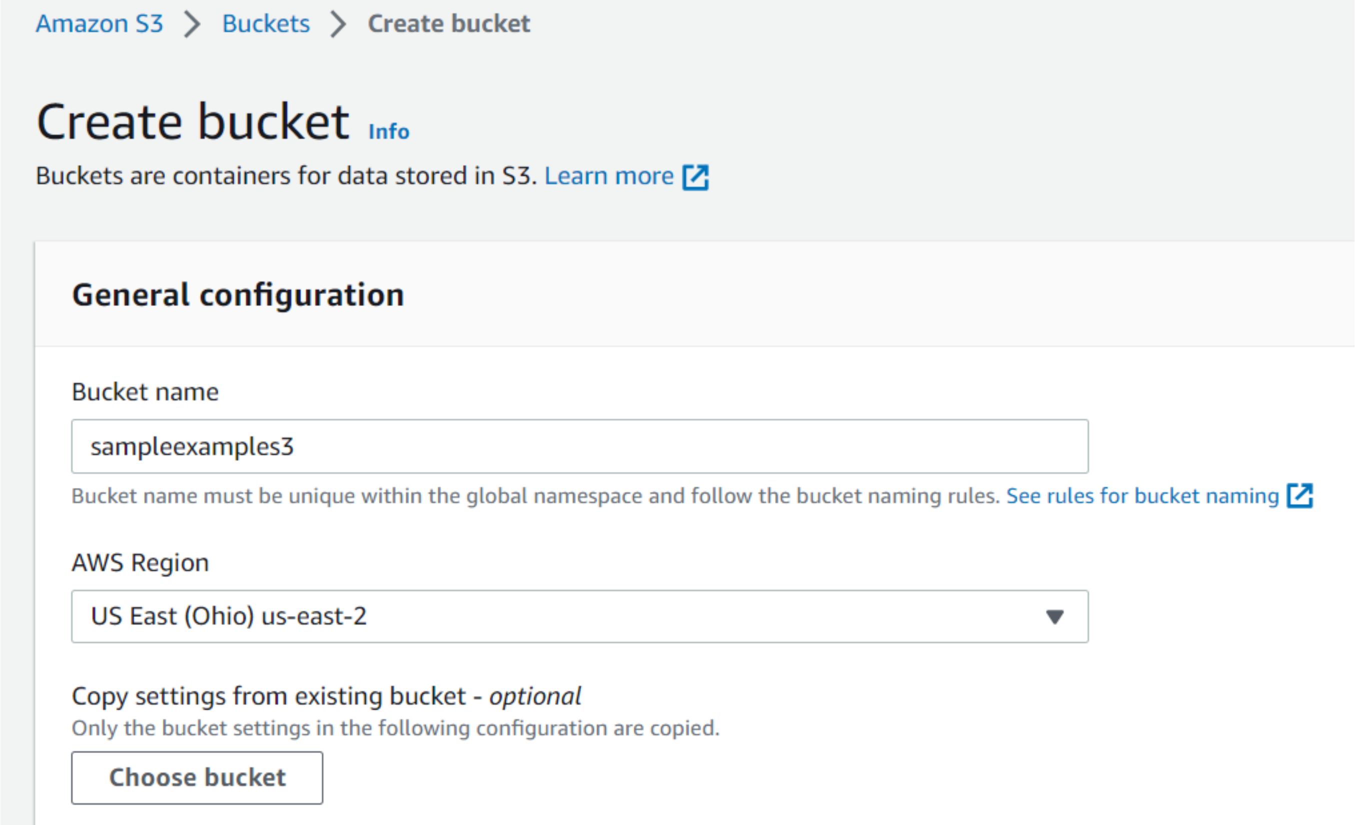


4. Set Bucket Properties:

- Bucket Name: Provide a distinctive name for your bucket, keeping in mind that bucket names must be unique globally across AWS.

- Region: Choose the desired AWS region where you wish to create the bucket.

- Adjust options as necessary: Enable or disable features such as versioning, server access logging, and default encryption based on your requirements.



Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration

Bucket name

sampleexamples3

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

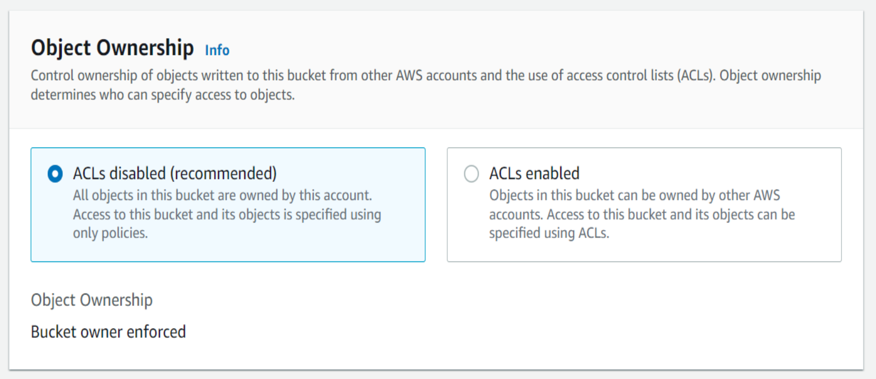
AWS Region

US East (Ohio) us-east-2

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

- When ACL is enabled, it means the bucket owner can specify fine-grained access controls for individual objects within the bucket. This allows the owner to grant or deny access to specific users or groups.
- On the other hand, when ACL is disabled, it means that the bucket owner's permissions are applied to all objects within the bucket. In this case, the access control is inherited from the bucket level, and individual object-level access control is not available.



Object Ownership Info

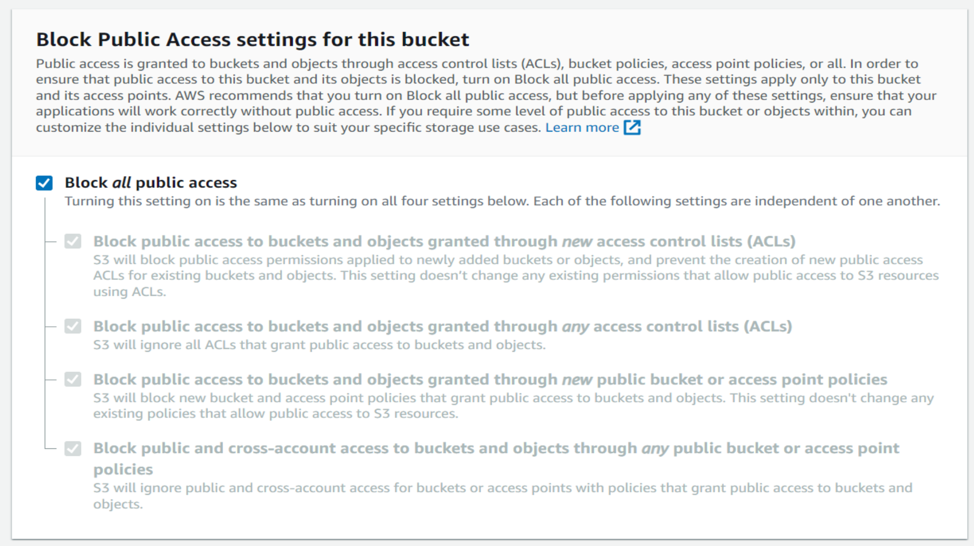
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

- This option lets you configure settings to prevent public access to the bucket and its objects. You can choose to block public access at the bucket level or apply stricter settings at the object level.



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

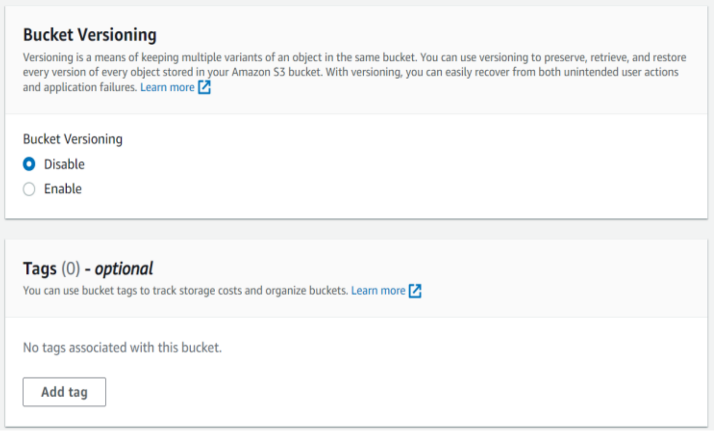
☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.

☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

- Enabling bucket versioning allows you to store multiple versions of objects within the bucket. This feature provides additional data protection and gives you the ability to track changes and revert to previous versions if needed.



Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

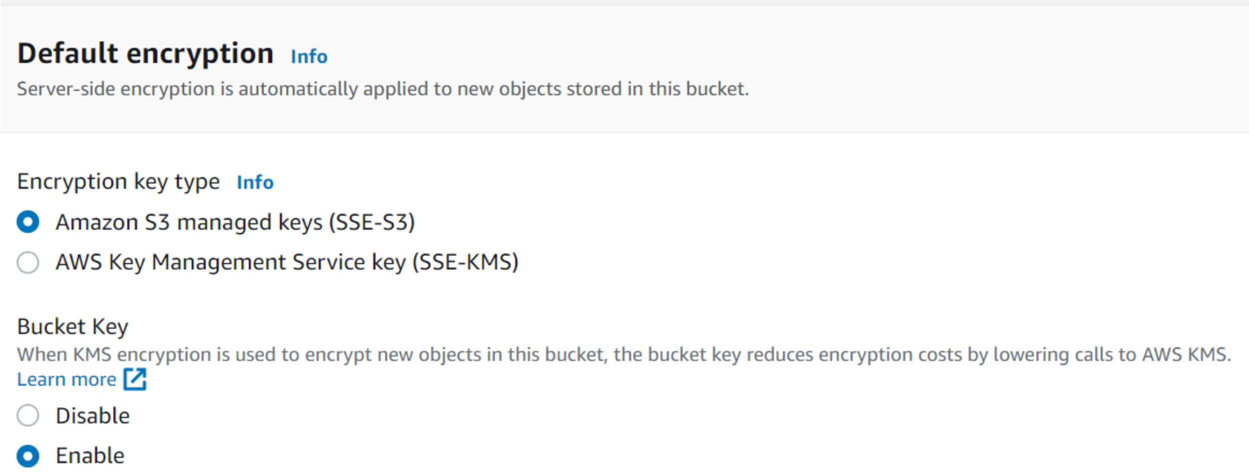
Bucket Versioning
● Disable
○ Enable

Tags (0) - ***optional***

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

- SSE-S3 is a server-side encryption option provided by AWS S3. When SSE-S3 is enabled for a bucket, S3 automatically encrypts the objects at rest using its own managed keys. The encryption and decryption processes are transparent to you, and you don't need to manage the encryption keys explicitly.
- With SSE-KMS, S3 encrypts the objects at rest using AWS Key Management Service (KMS) keys. KMS provides a highly secure and scalable key management solution. SSE-KMS allows you to have more control over the encryption process by using customer-managed keys (CMKs) provided by AWS KMS.



Default encryption Info

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

Encryption key type Info

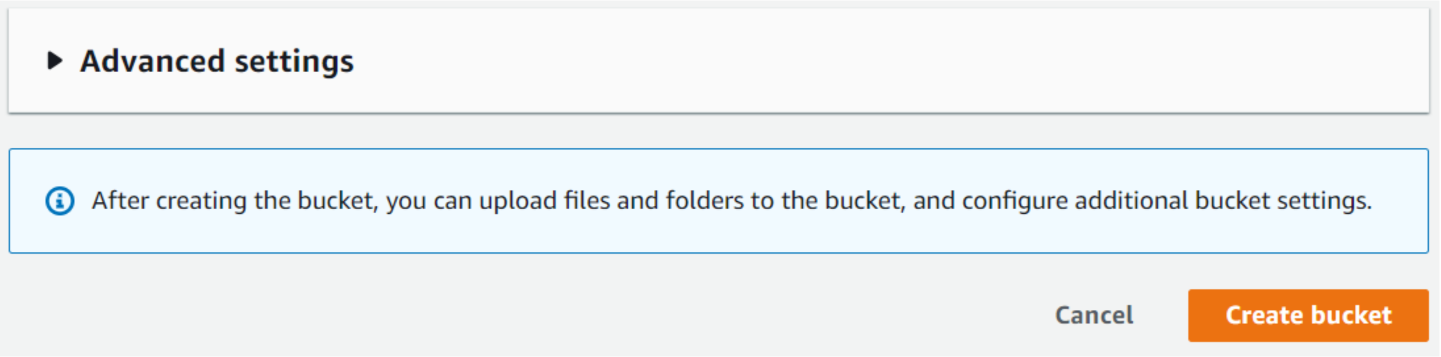
● Amazon S3 managed keys (SSE-S3)
○ AWS Key Management Service key (SSE-KMS)

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

○ Disable
● Enable

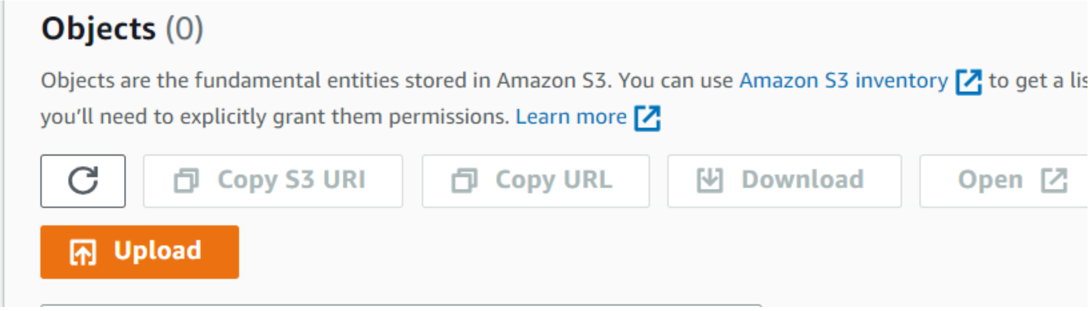
- Now, we can click on create bucket.



▶ **Advanced settings**

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

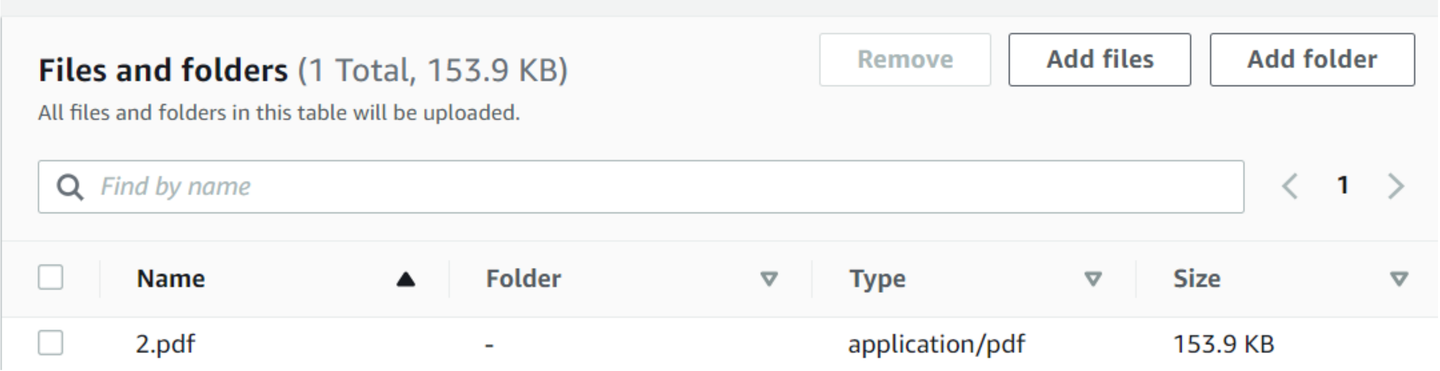
- After successfully creating the bucket, you have the ability to create directories and transfer files from your local machine to the cloud. This can be accomplished by utilizing the upload icon located within the bucket interface.
- We can upload the data using the upload button.



Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a lis
you'll need to explicitly grant them permissions. Learn more

- We can just drag and drop the files or can also upload using the add files option.

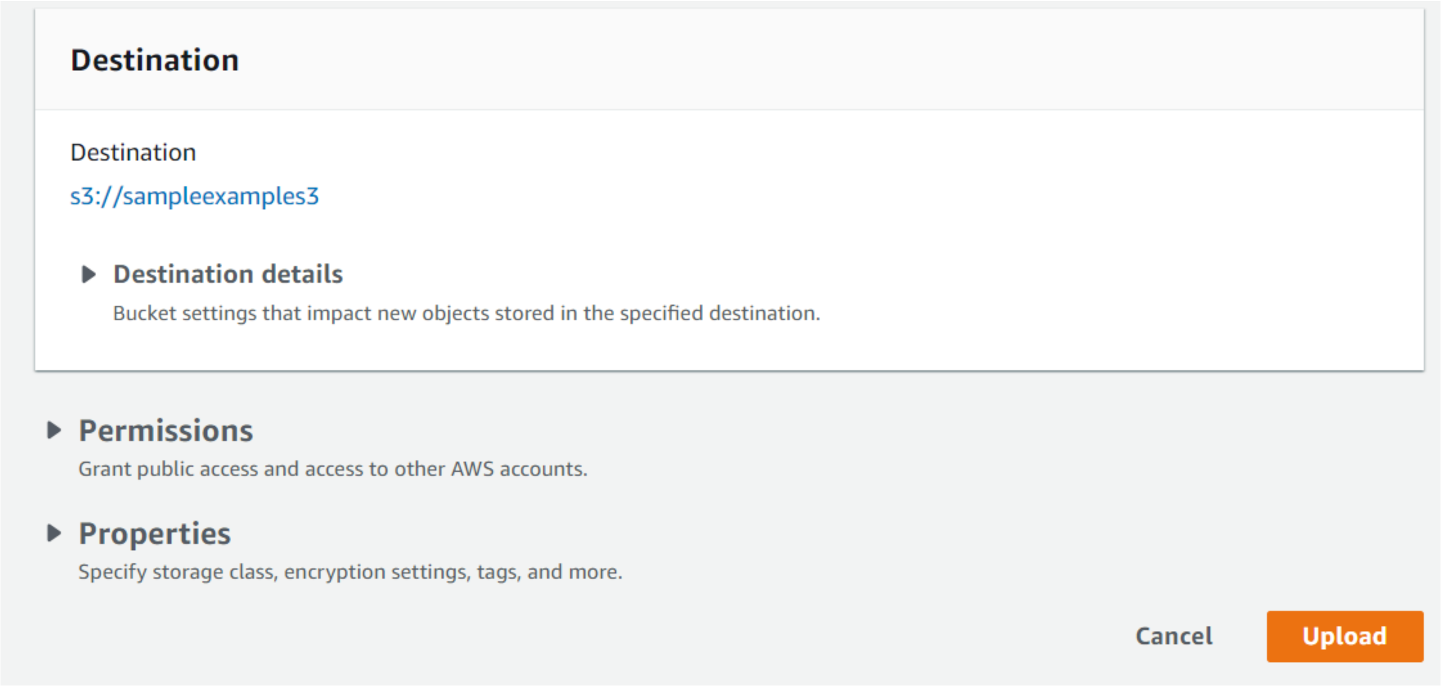


Files and folders (1 Total, 153.9 KB)

All files and folders in this table will be uploaded.

□	Name ▲	Folder ▽	Type ▽	Size ▽
□	2.pdf	-	application/pdf	153.9 KB

- Under the permissions tab, we can grant the public access to other AWS accounts, and under the properties you have different storage types that you can select for your file storage, and then click on the upload button.



Destination

Destination
s3://sampleexamples3

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

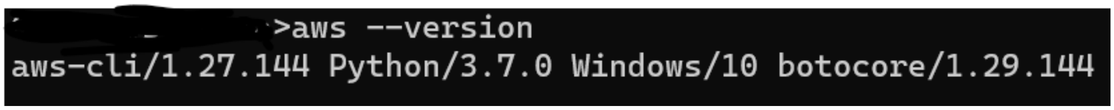
▶ **Permissions**
Grant public access and access to other AWS accounts.

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

- We can delete, copy, and download the uploaded file by selecting the check box next to the file and then choosing the necessary actions that have to be done.

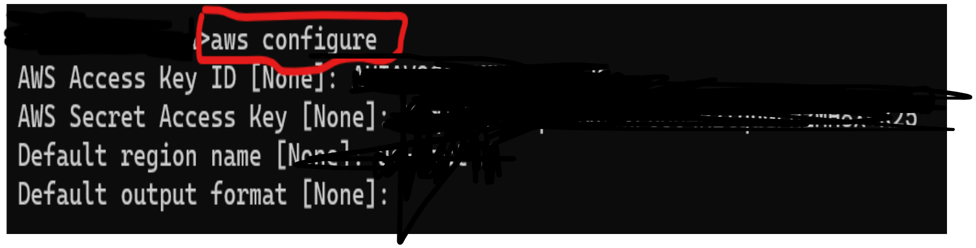
Creating an S3 Bucket using the AWS CLI:

- Make sure to have the AWS CLI installed on your local computer by following the installation instructions provided by AWS. To verify the installation, you can use the command "**aws --version**" in the command prompt.



```
[redacted]>aws --version
aws-cli/1.27.144 Python/3.7.0 Windows/10 botocore/1.29.144
```

- To configure the AWS CLI, you can run the command "aws configure" in your command prompt or terminal. This will prompt you to enter your AWS Access Key ID, Secret Access Key, default region, and output format. By providing these credentials and settings, the CLI will be properly configured to interact with your AWS resources.

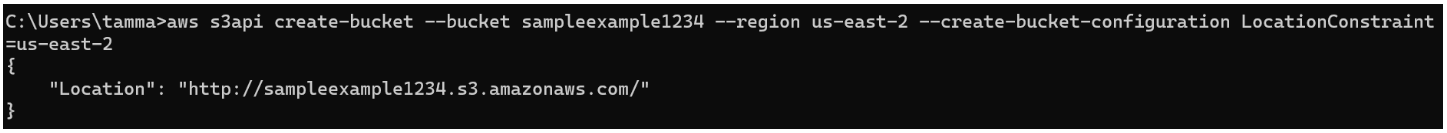


```
[redacted]>aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: [redacted]
Default output format [None]:
```

- To create a bucket, use the following command:

  aws s3api create-bucket --bucket your-bucket-name --region your-region –create-bucket-configuration LocationConstraint=your-region

  Replace **your-bucket-name** with your desired bucket name, and **your-region** with your preferred AWS region.



```
C:\Users\tamma>aws s3api create-bucket --bucket sampleexample1234 --region us-east-2 --create-bucket-configuration LocationConstraint
=us-east-2
{
    "Location": "http://sampleexample1234.s3.amazonaws.com/"
}
```

- You can validate the bucket creation by accessing the AWS Management Console and confirming the presence of the newly created bucket in the interface.
- We can upload the files on the local machine using the below command:

  aws s3 cp your-local-machine-file-path s3://your-bucketname-key
- We can verify on the console if the file is uploaded or not by navigating into the bucket.
- We can also remove the files using the rm command as follows, and check with the same on the console.

  aws s3 rm s3://bucket-name/key-name