Configuring Amazon S3

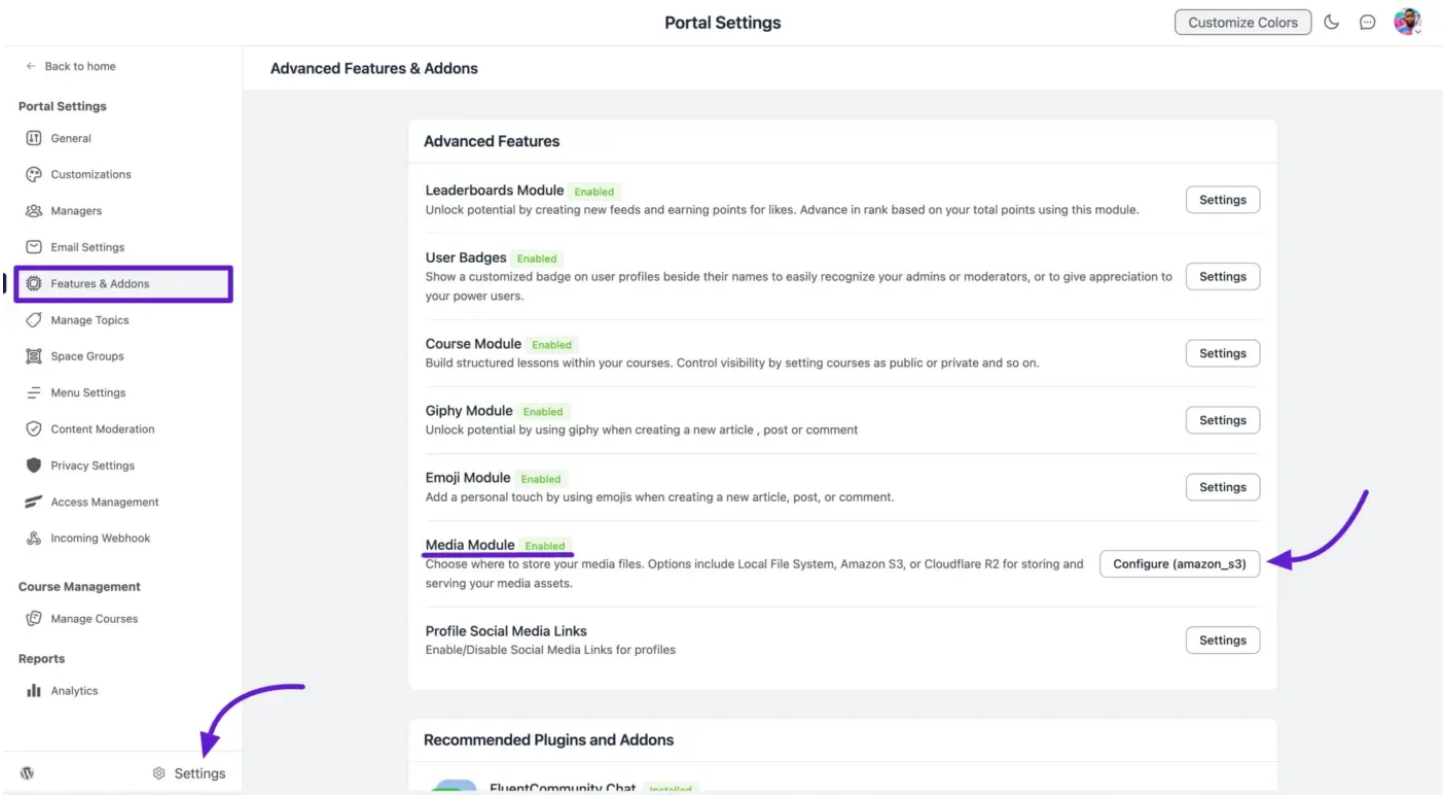
Integration · https://fluentcommunity.co/docs/configuring-amazon-s3/

You can integrate your Amazon S3 bucket with FluentCommunity to store media files externally, reducing your site's load and maintaining its storage health.

This guide will walk you through the easy steps to set up Amazon S3 with FluentCommunity.

Steps to Configure Amazon S3

First, navigate to the **Settings** of your FluentCommunity. From the left sidebar, click on **Features & Addons**, then select **Media Module Settings**.

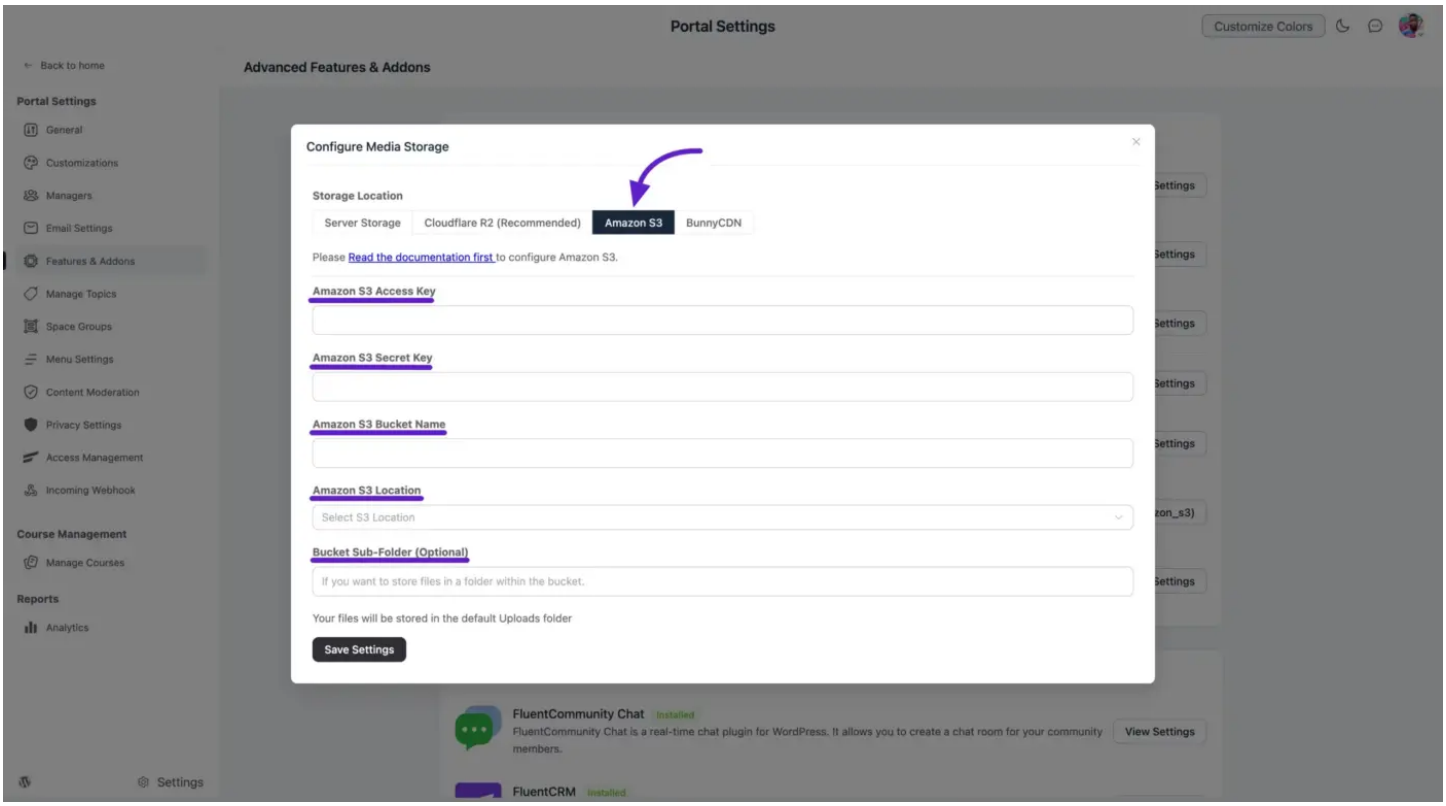


Next, you'll see all the available cloud storage options. Click on **Amazon S3** from the menu, and the required fields to connect FluentCommunity with Amazon S3 will appear.

To set up the connection, you'll need the following credentials:

- **Amazon S3 Access Key**
- **Amazon S3 Secret Key**
- **Amazon S3 Bucket Name**
- **Amazon S3 Location**
- **Bucket Sub Folder** (Optional)

You can find these credentials in your Amazon S3 account. In the next section, we'll walk you through how to retrieve them step by step.

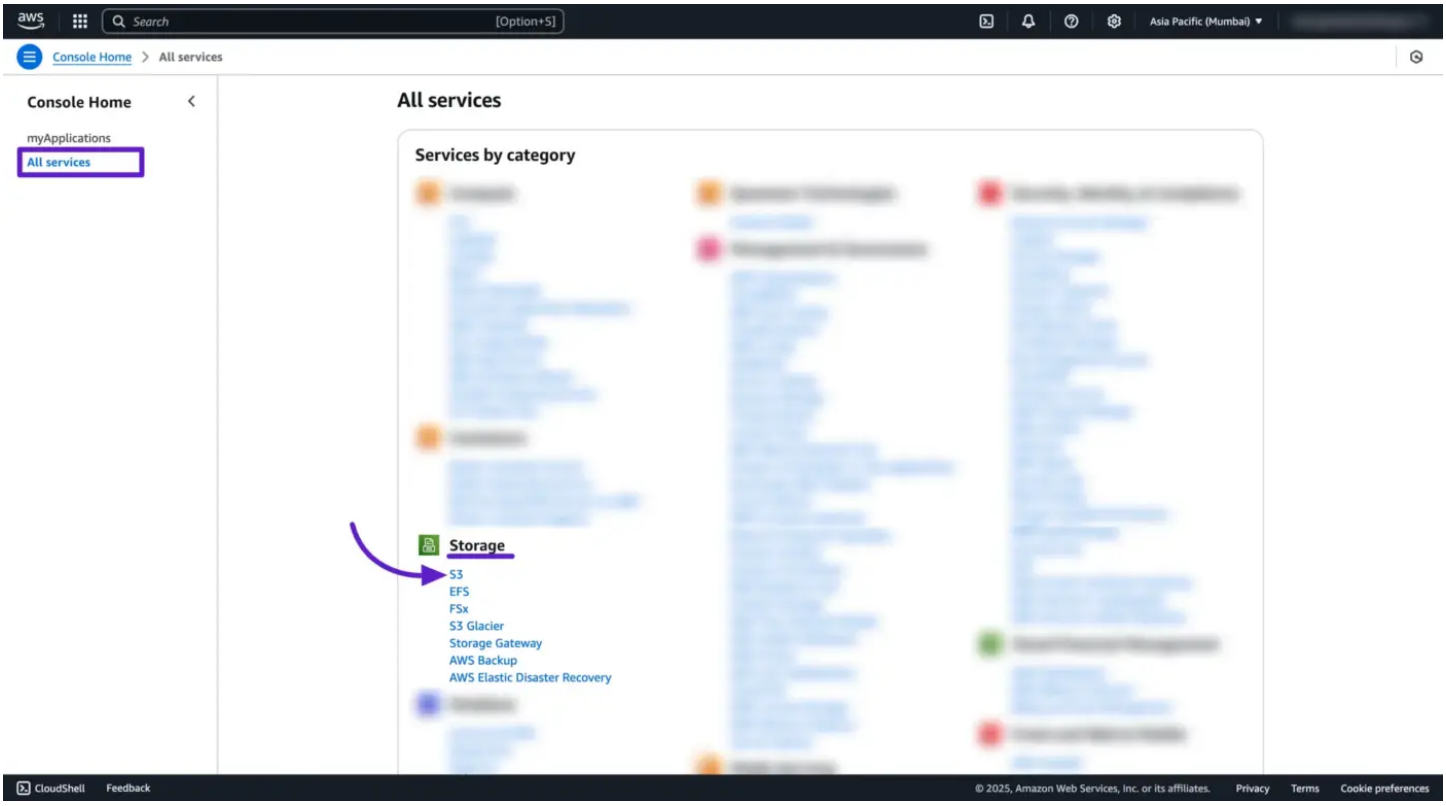


Get the Credentials

Creating S3 Bucket

Log in to your **Amazon AWS** account to get the necessary credentials.

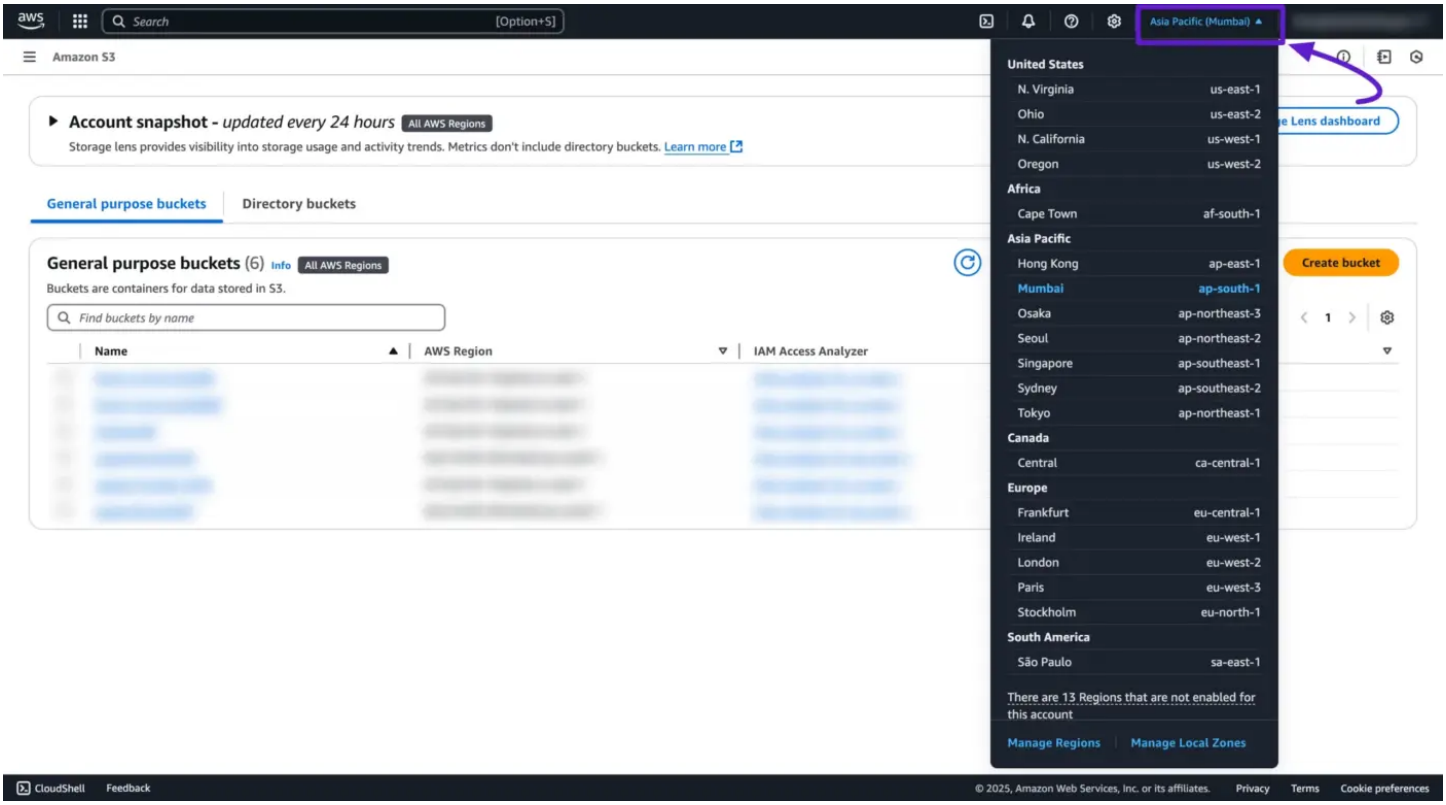
From the **Dashboard**, click on **All Services** in the left sidebar. A full list of AWS services will appear—scroll down to the **Storage** section and select **S3** to proceed.



Before creating your bucket, you need to select a region. Amazon S3 creates each bucket in the region you choose, which can help optimize performance, reduce latency, lower costs, or meet data compliance requirements.

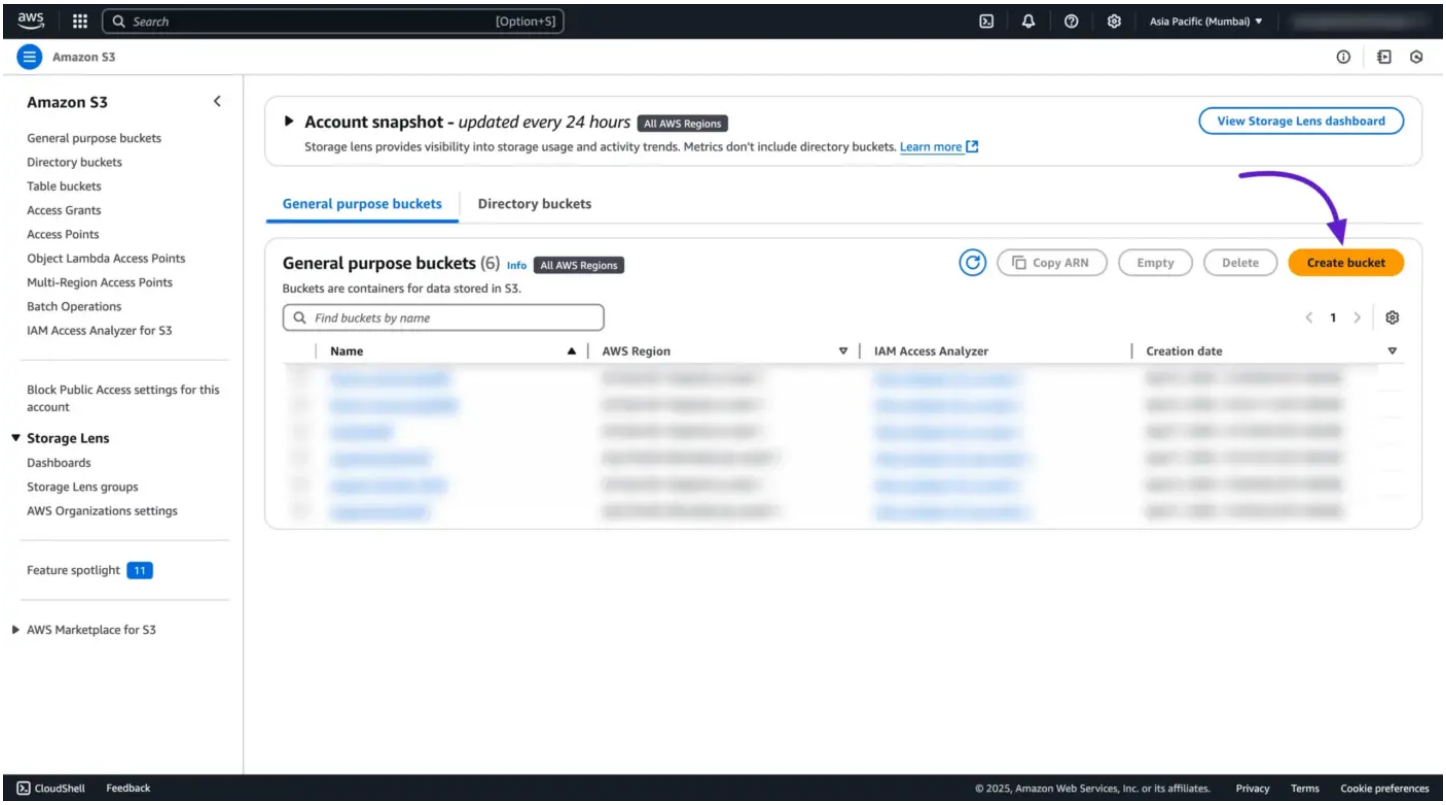
For instance, if you're based in Europe, it's often best to choose a region like **EU (Ireland)** or **EU (Frankfurt)**.

To select a region, click the **Region name** at the top of your AWS console and choose your preferred region from the dropdown list. For a full list of available AWS S3 regions, check this Regions and Endpoints.



Now it's time to create a bucket where all your FluentCommunity media files will be stored.

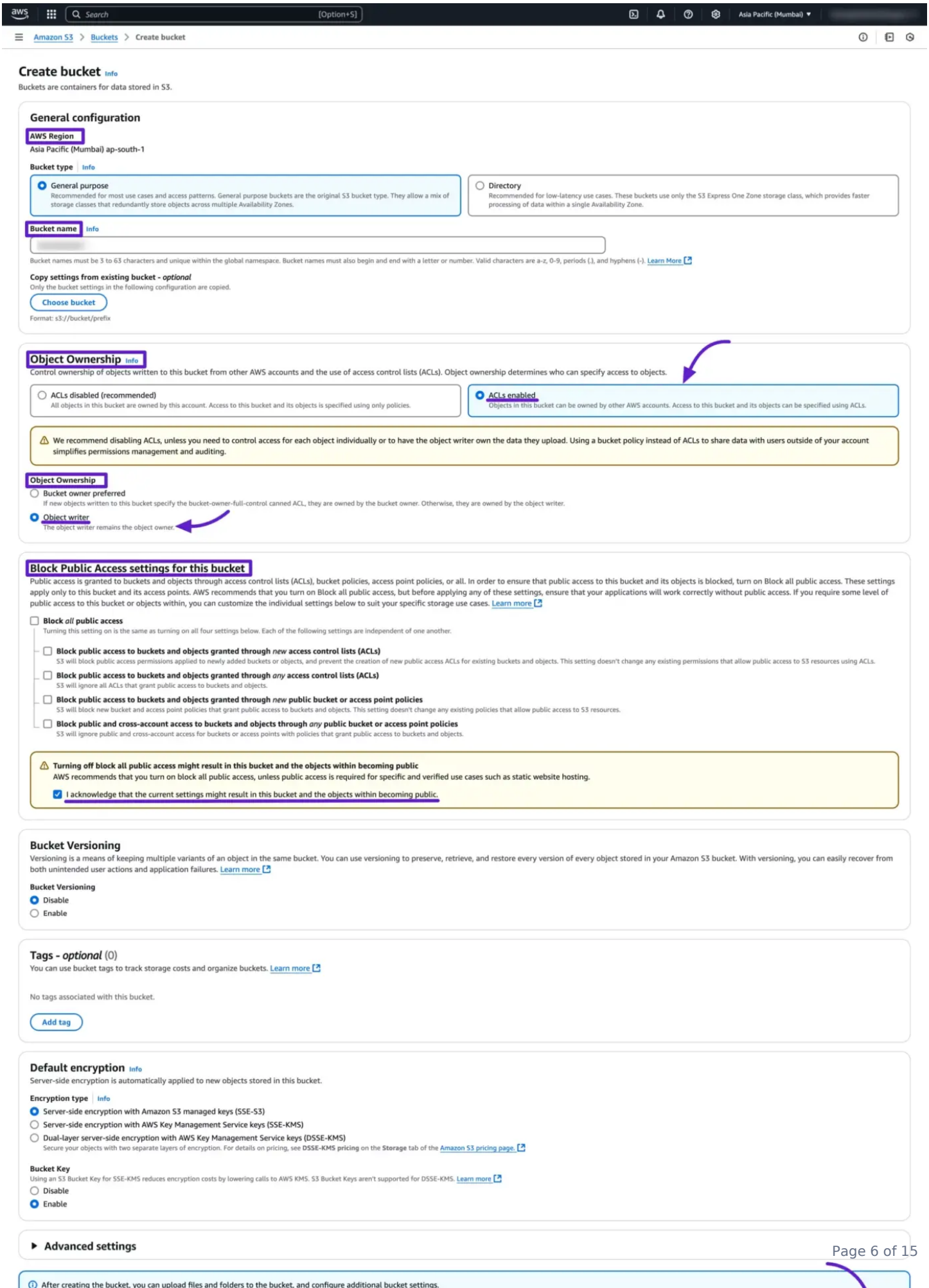
To do this, simply click the **Create Bucket** button. This will start the setup process for your new storage bucket.



You'll now be redirected to the bucket creation page. Here's how to fill out the form and set the correct permissions:

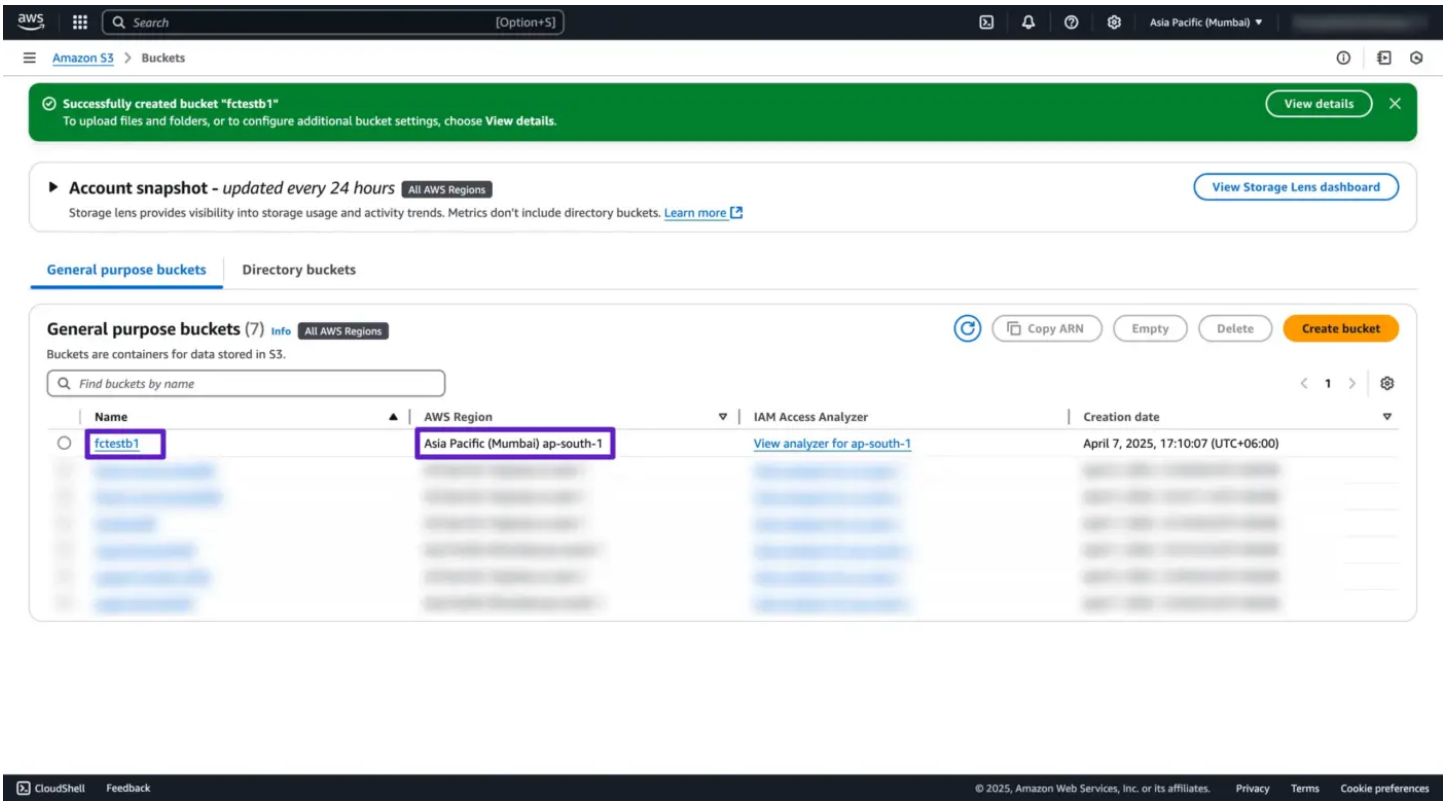
- **AWS Region**: Make sure the region matches the one you selected earlier.
- **Bucket Name**: Enter a unique name for your bucket.
- **Object Ownership**:
  - Select **ACLs enabled**.
  - Then choose **Object writer** under Object Ownership.
- **Block Public Access Settings for this Bucket**:
  - Disable **Block All Public Access** permissions.
  - Check the box saying **I acknowledge that the current settings might result in this bucket and the objects within becoming public.**
- **Other Settings**: You can customize them based on your needs or keep the defaults.

Finally, click the **Create Bucket** button to complete the process.


Search [Option+S]
Asia Pacific (Mumbai)
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
Asia Pacific (Mumbai) ap-south-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
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.
We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.
Object Ownership
Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.
Object writer
The object writer remains the object owner.
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.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.
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 - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
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)
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.
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.

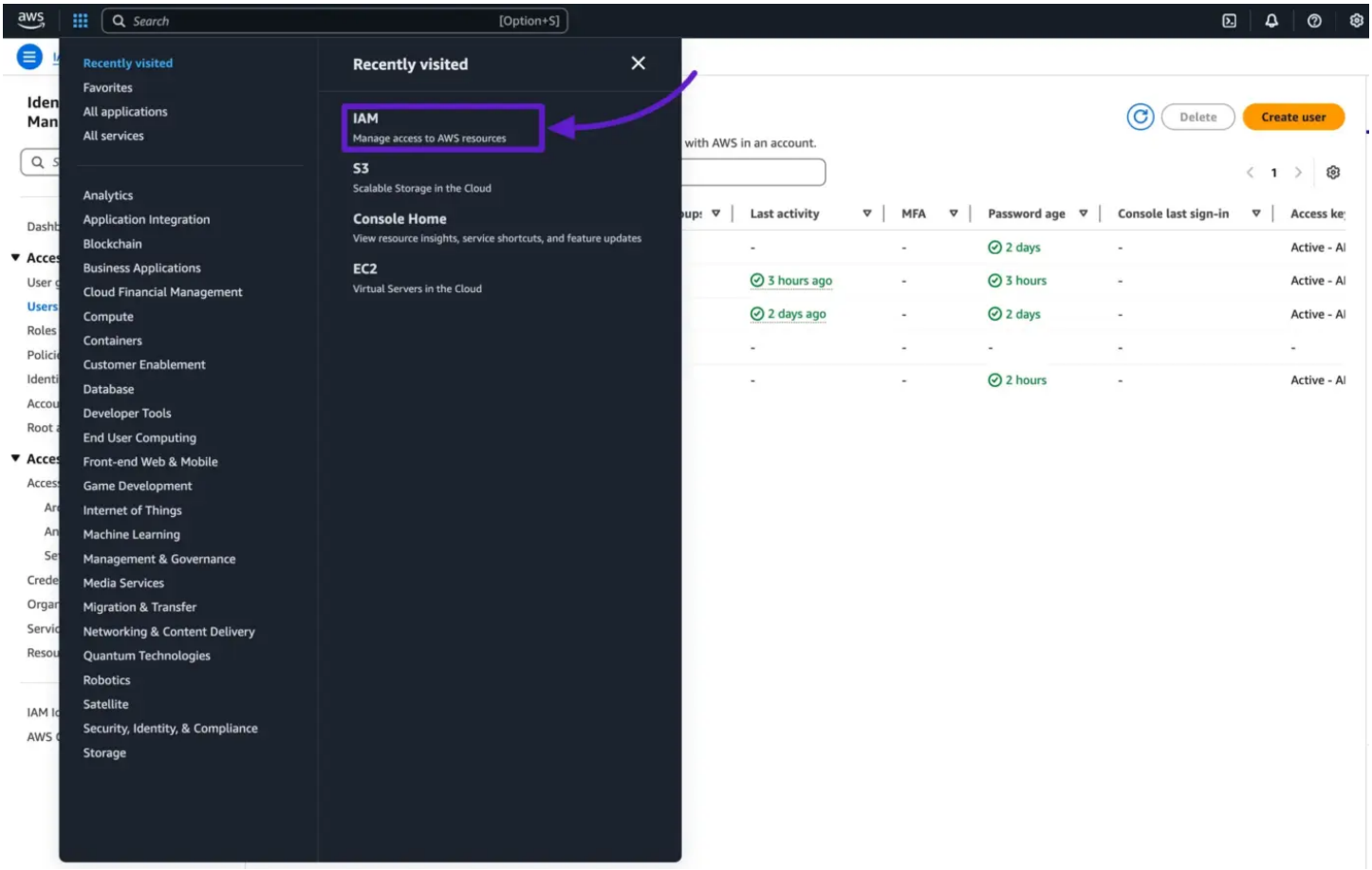
Once your bucket is successfully created, you'll see a confirmation message, and the bucket will appear in your list.

> Make sure to copy the **Bucket Name** and **Region**—you'll need these to fill in the **Bucket Name** and **Location** fields in your FluentCommunity settings.

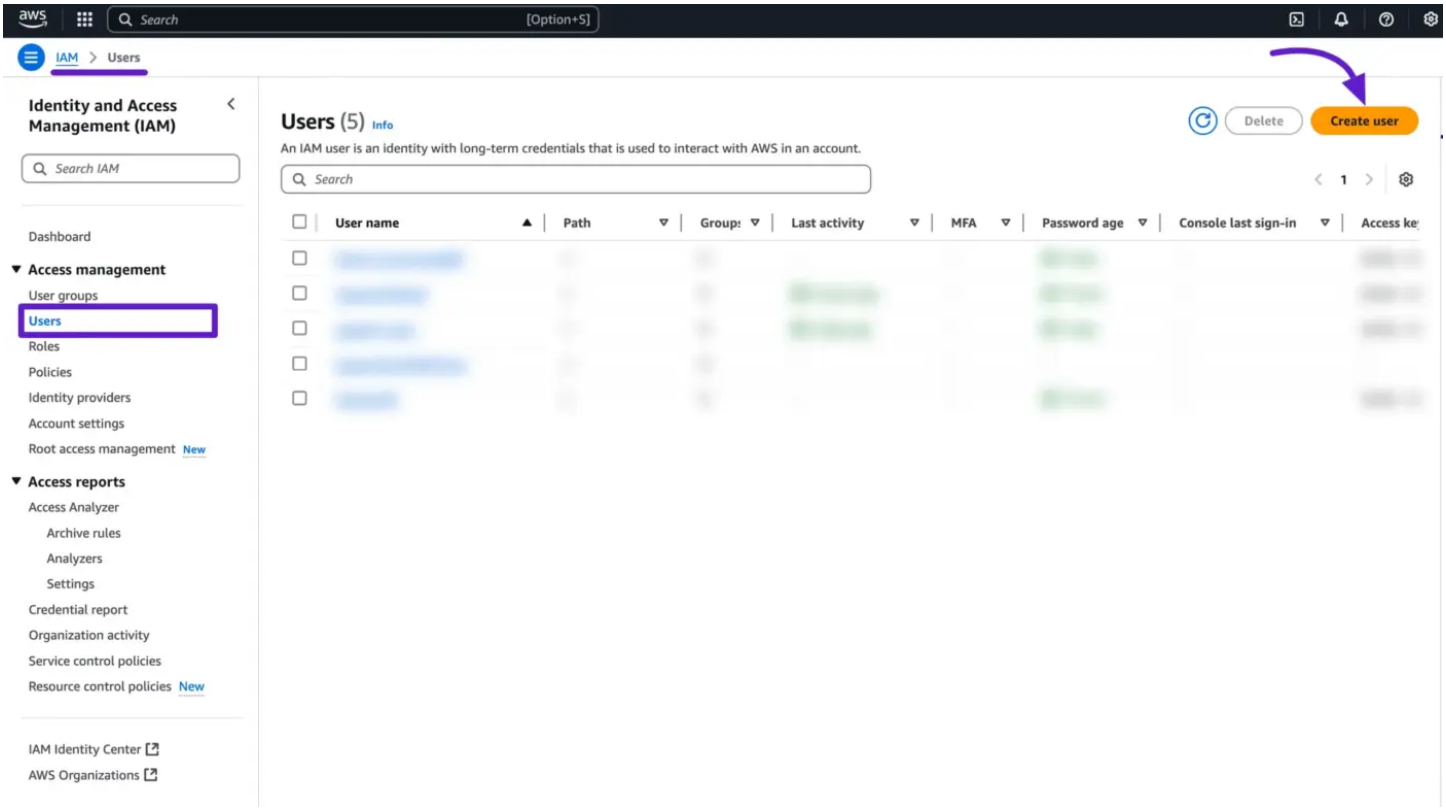


Creating IAM User for Access

Click the **Services** dropdown in the AWS navbar or search for **IAM**. Select **IAM** to begin creating a new IAM user for access.



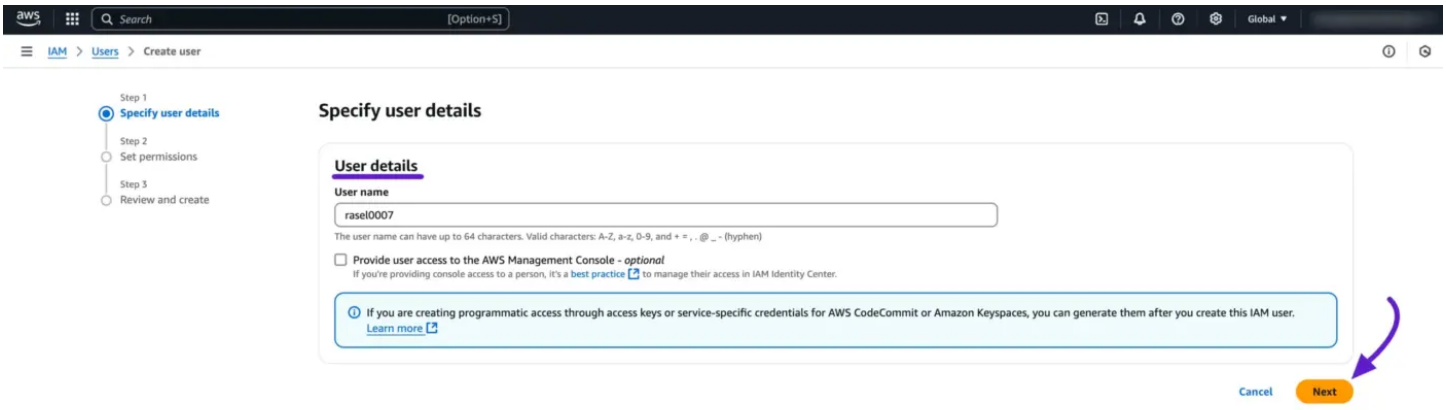
Once you're on the IAM dashboard, click **Users** from the left sidebar. Then hit the **Create User** button to start setting up a new IAM user.



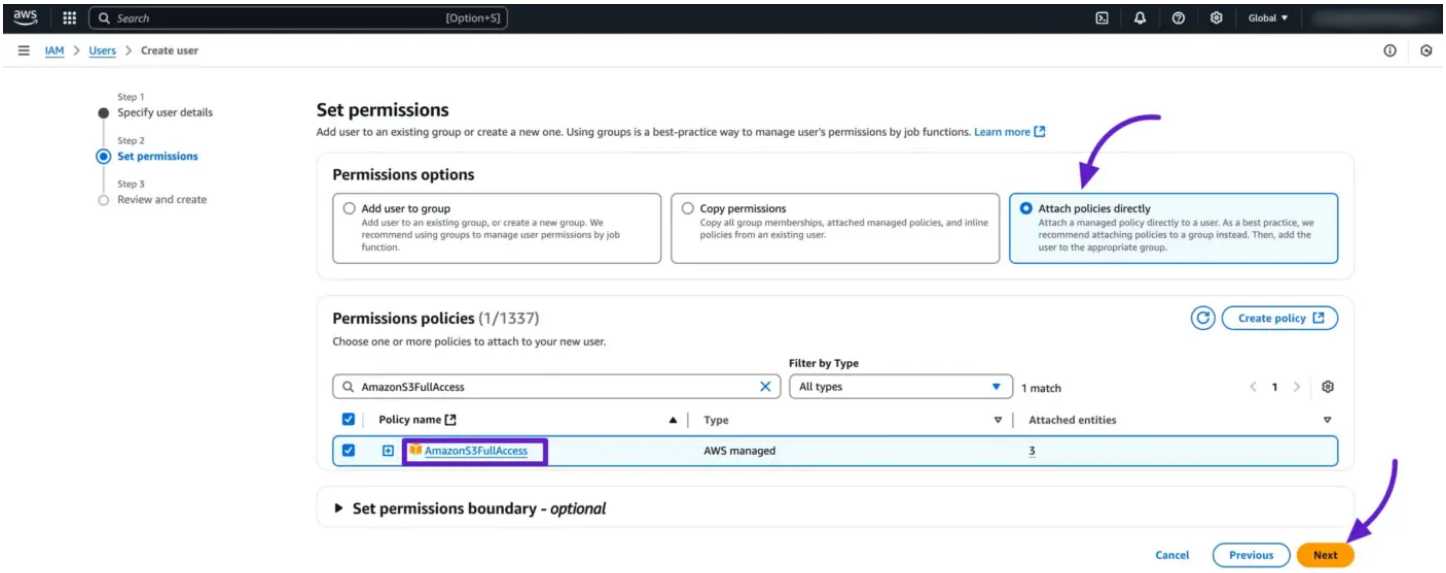
On the **Create User** page, start by entering your desired **User name**.

Next, leave the **Provide user access to the AWS Management Console (optional)** option unchecked, as it's not needed for FluentCommunity integration.

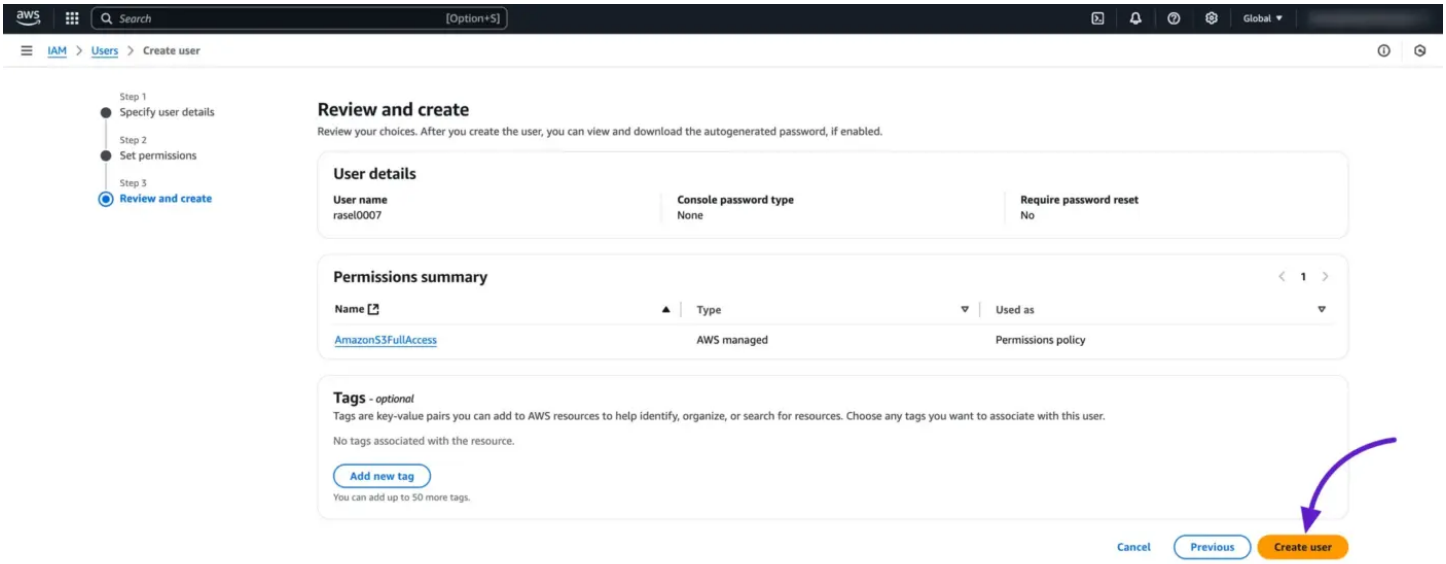
Then, click the **Next** button to proceed to the permissions setup.



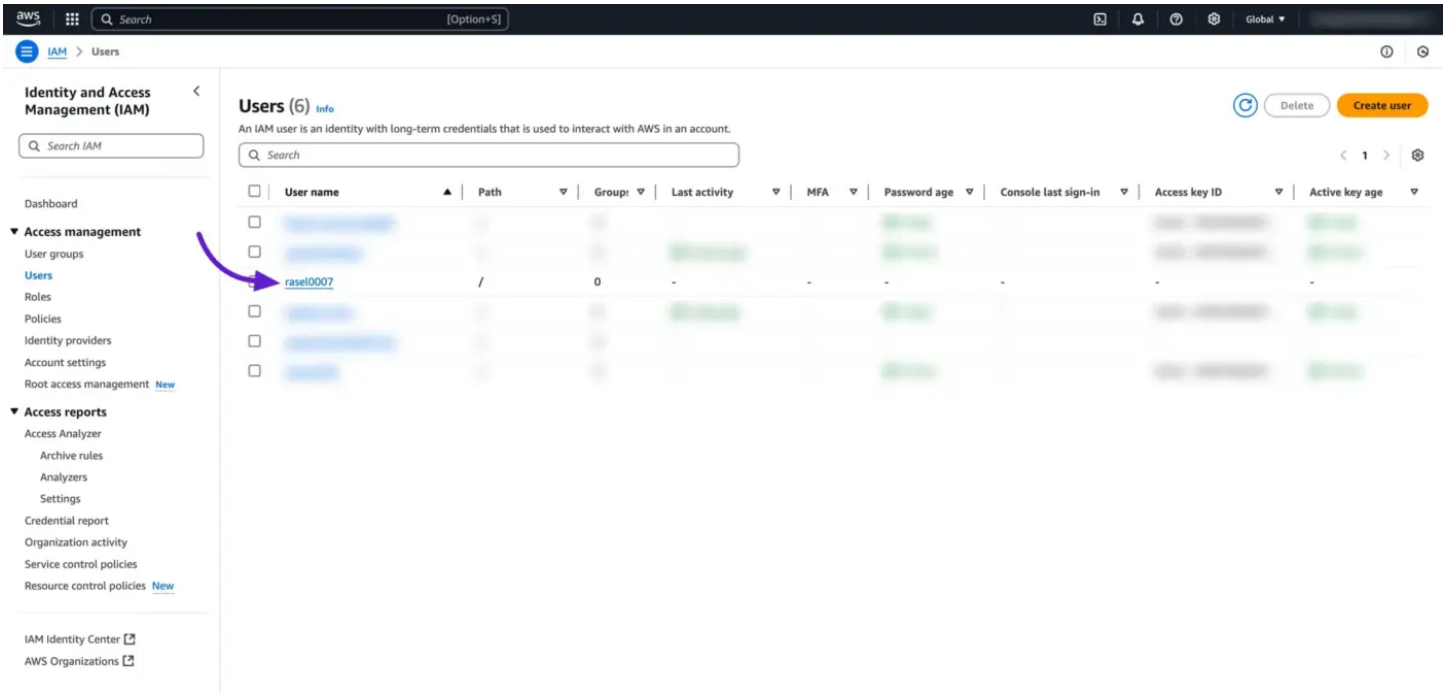
In the **Set Permissions** step, select **Attach policies directly**. Then, search for **AmazonS3FullAccess** in the search bar, select the policy, and click the **Next** button to proceed.



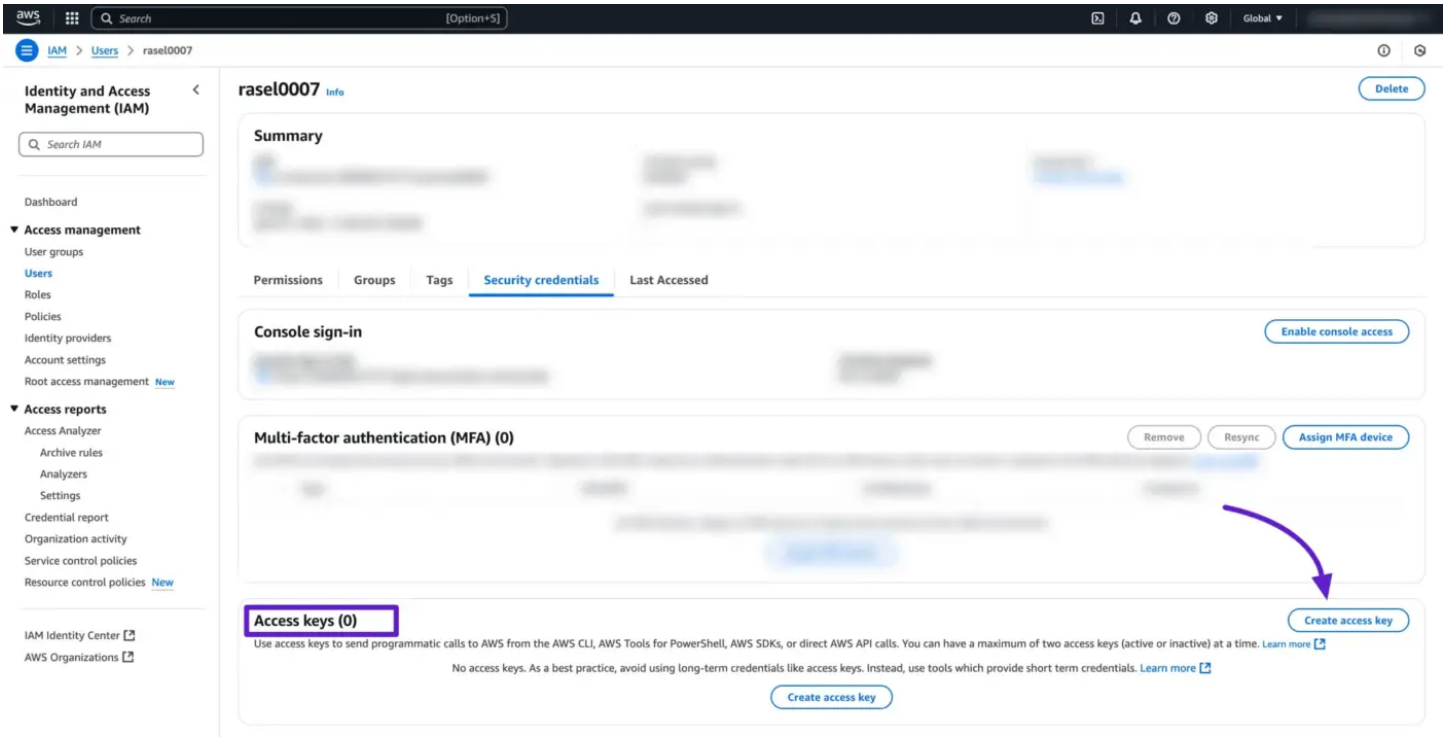
You can skip the **Review and Create** step and click on the **Create User** button to create the new user.



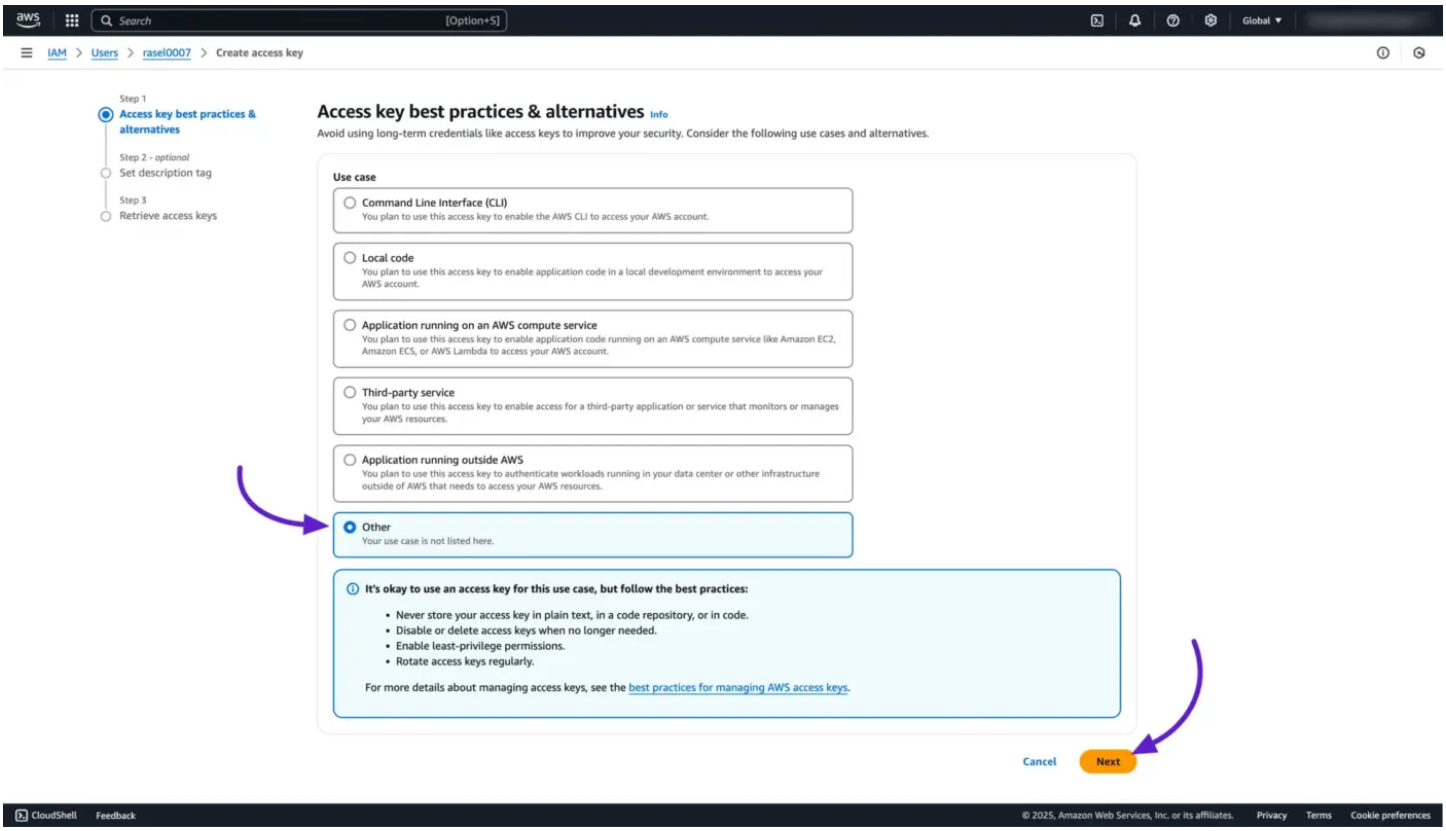
Your user is created successfully. In the **Users** section, you will now see the newly created user. Click on the **User Name** to view the details.



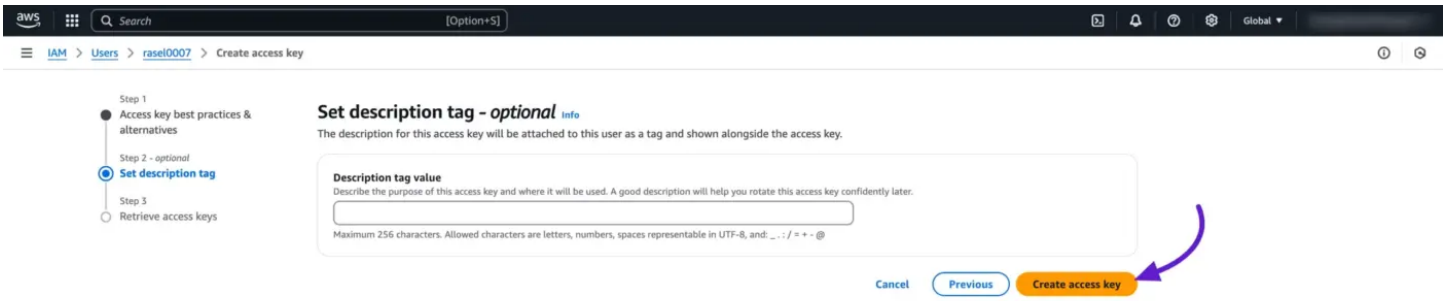
Now, go to the **Security Credentials** tab from the User Details page. Under this tab, you will find the **Access Keys** section. Click on the **Create Access Key** button to generate new access credentials for the user.



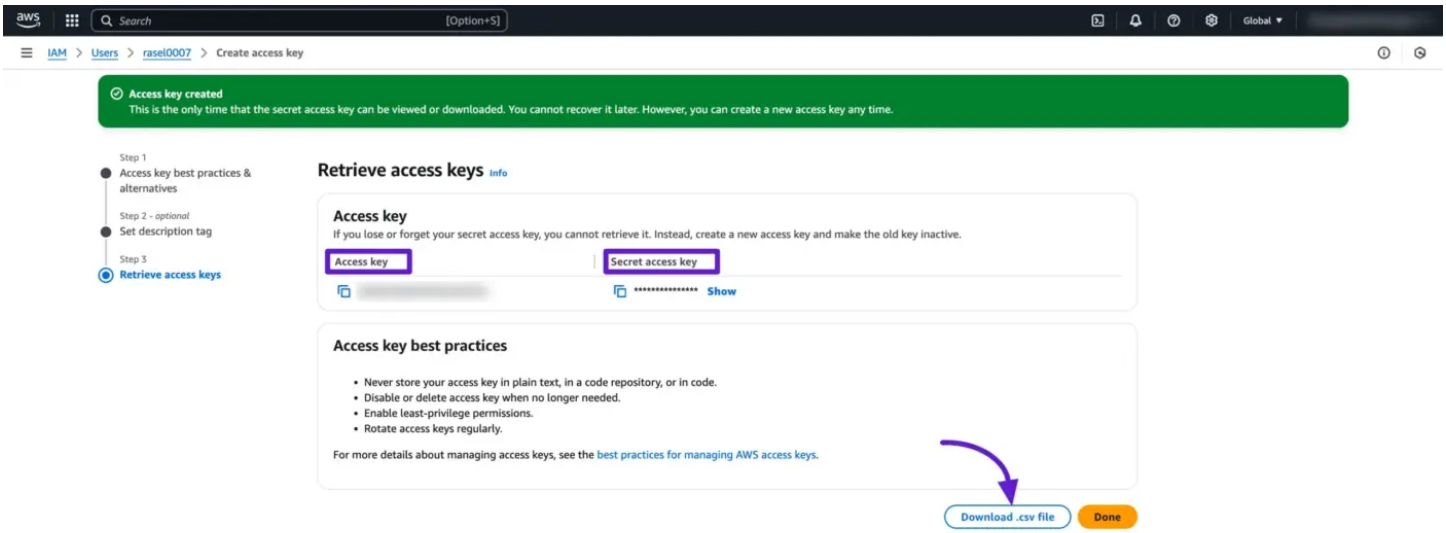
The steps to create Access Keys will appear. In the **Access Key Best Practices & Alternatives** step, select the **Other** option and click on the **Next** button to proceed.



You can simply skip this step and click on the **Create Access Key** button to generate the access key.



Your Access Keys are created now. You can see the Access Key and Secret Key here. Copy these credentials to your clipboard for later use or download them as a CSV file by clicking on the **Download CSV File** button.

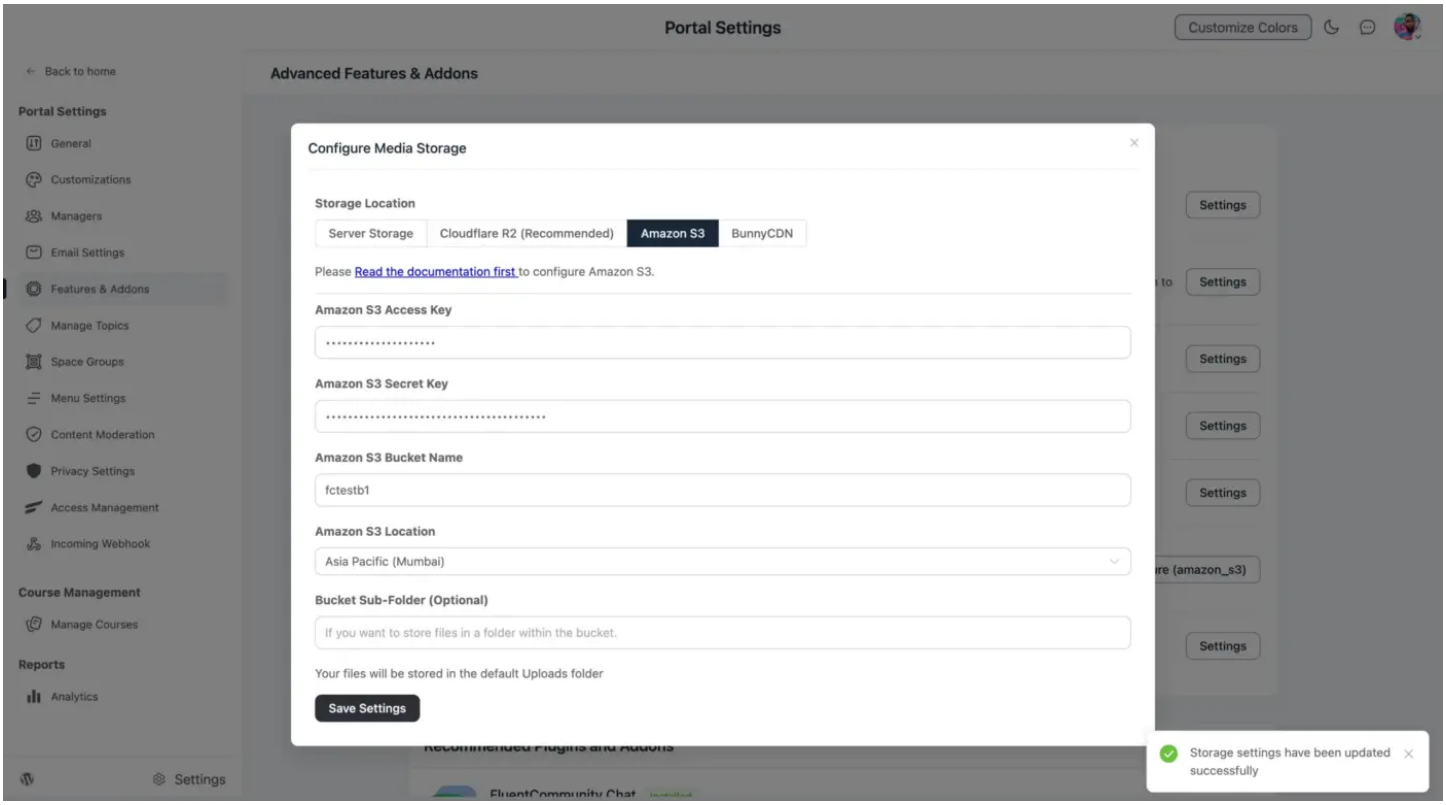


Configuring Amazon S3

Now, go to the FluentCommunity Media Storage Settings and select **Amazon S3**. Paste the credentials you collected from your Amazon AWS account:

- **Amazon S3 Access Key**: Paste the Access Key from the IAM User.
- **Amazon S3 Secret Key**: Paste the Secret Key from the IAM User.
- **Amazon S3 Bucket Name**: Enter the Bucket Name you created in Amazon S3.
- **Amazon S3 Location**: Select the region from the dropdown that matches the one you used when creating your bucket.
- **Bucket Sub Folder (Optional)**: If you created a subfolder for file uploads, provide its name. Otherwise, leave it empty.

Finally, click the **Save Settings** button. Your Amazon S3 will now be successfully connected with FluentCommunity, and all media files will be stored in Amazon S3.



Configure Amazon S3 Using wp-config.php

Add the following definitions to your `wp-config.php` file, adjusting the values according to your Amazon S3 setup.

```
// Amazon S3 Configuration
define('FLUENT_COMMUNITY_CLOUD_STORAGE', 'amazon_s3');
define('FLUENT_COMMUNITY_CLOUD_STORAGE_REGION', 'us-east-1'); // change with your region. If it's global just remove this line or keep it empty
define('FLUENT_COMMUNITY_CLOUD_STORAGE_ACCESS_KEY', '********************');
define('FLUENT_COMMUNITY_CLOUD_STORAGE_SECRET_KEY', '********************');
define('FLUENT_COMMUNITY_CLOUD_STORAGE_BUCKET', 'your-bucket-name'); // change with your bucket name
define('FLUENT_COMMUNITY_CLOUD_STORAGE_SUB_FOLDER', 'site-name'); // optional. If you want to store the files in a subfolder of that bucket
```

Replace the placeholder values with your actual Amazon S3 credentials and information.

Notes

- If you use both methods, the wp-config.php settings will take precedence over the UI settings.
- The UI method is more user-friendly and doesn't require direct file access to your WordPress installation.
- The wp-config.php method is more secure as it keeps sensitive information out of the database.
- Ensure that all the credentials are correctly entered, regardless of the method you choose.
- The sub-folder setting is optional and can be used if you want to organize your files within the bucket.

Troubleshooting

- If you encounter permission issues, review your bucket policy and IAM user permissions.
- Ensure that your S3 bucket is in the correct region and that it's accessible from your WordPress server.
- Check that your access key and secret key are entered correctly without any extra spaces.

If you have any questions regarding this guideline, please don’t hesitate to reach out to our support team. We're here to assist you!