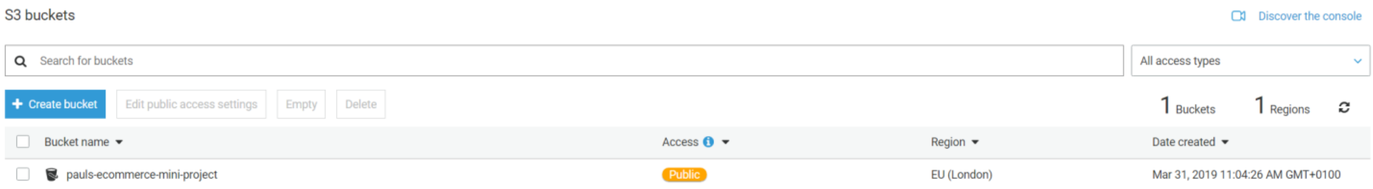
Create new S3 Bucket

I am creating a new S3 Bucket in AWS for Static website hosting for my Full stack Frameworks for Django project no5.

- Log into your AWS account https://aws.amazon.com/console/
- Go to S3 buckets which allows you to create buckets which are small parts of a cloud server that you can put your files into.


S3 buckets

Discover the console

Search for buckets

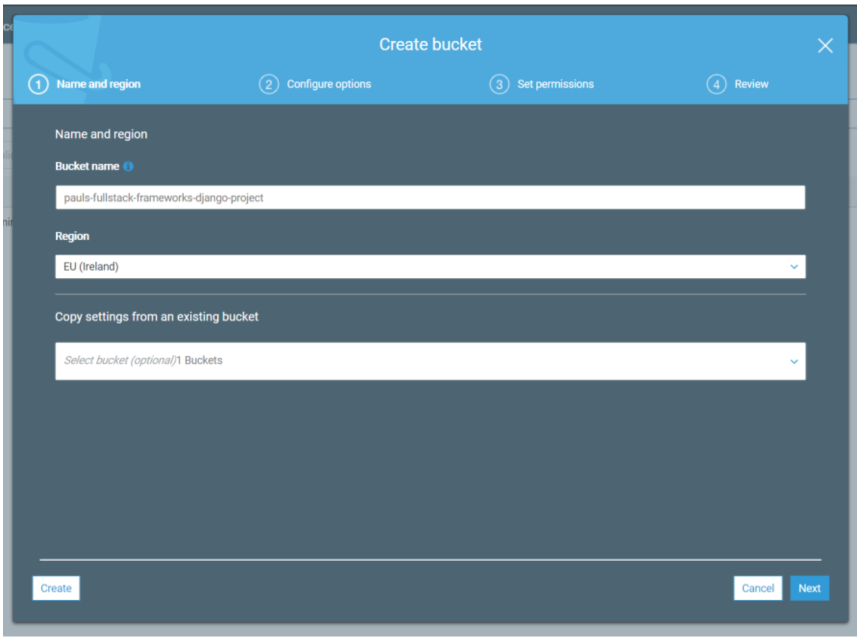
All access types

Create bucket | Edit public access settings | Empty | Delete

1 Buckets 1 Regions

Bucket name	Access	Region	Date created
pauls-ecommerce-mini-project	Public	EU (London)	Mar 31, 2019 11:04:26 AM GMT+0100

- Click on Create bucket button


Create bucket

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

Name and region

Bucket name: pauls-fullstack-frameworks-django-project

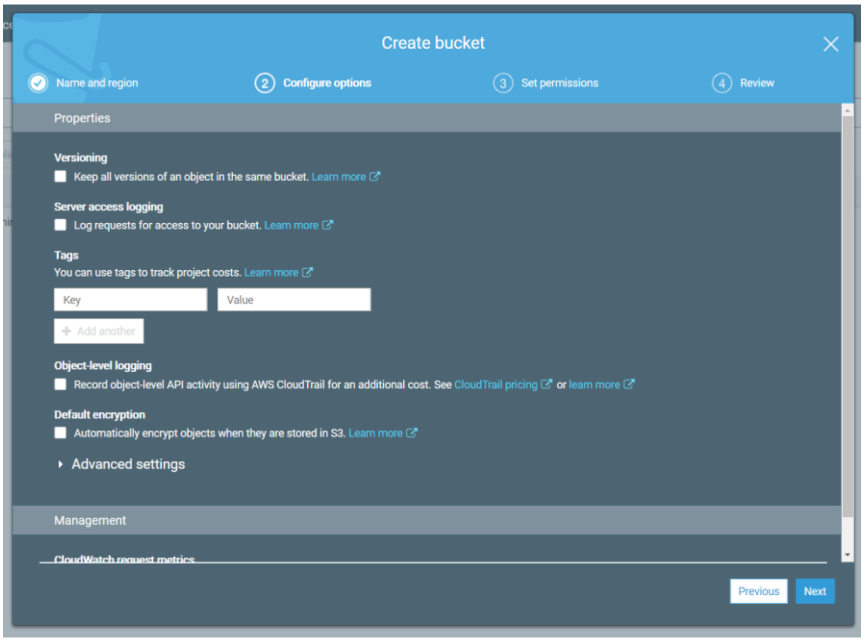
Region: EU (Ireland)

Copy settings from an existing bucket

Select bucket (optional)1 Buckets

Create | Cancel | Next

- Enter name of bucket in Bucket name field and then click on next button


Create bucket

Name and region 2 Configure options 3 Set permissions 4 Review

Properties

Versioning

□ Keep all versions of an object in the same bucket. Learn more

Server access logging

□ Log requests for access to your bucket. Learn more

Tags

You can use tags to track project costs. Learn more

Key | Value

+ 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

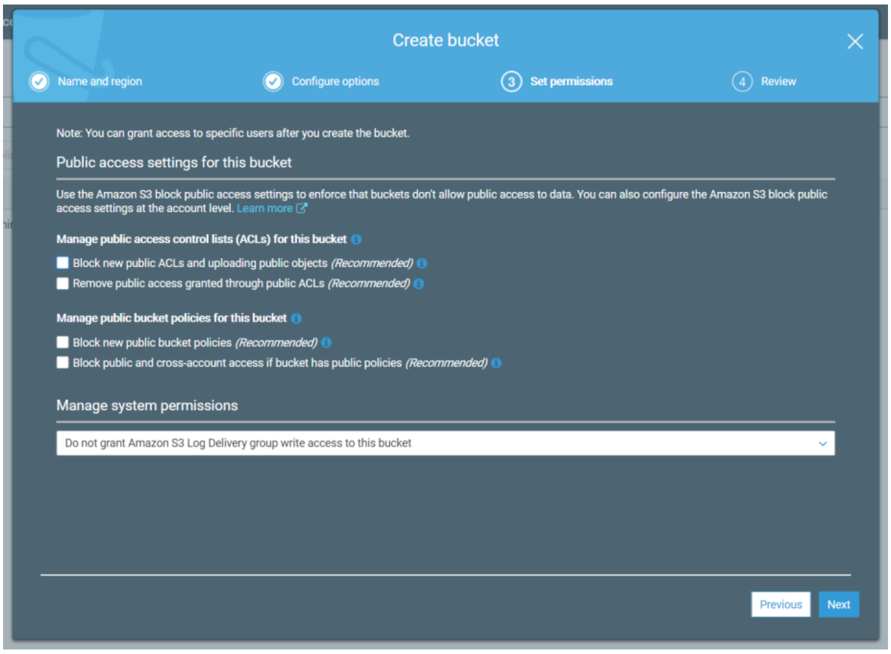
Advanced settings

Management

CloudWatch request metrics

Previous | Next

- Just click on the next button as nothing to fill in on this page.



Note: You can grant access to specific users after you create the bucket.

Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more

Manage public access control lists (ACLs) for this bucket

□ Block new public ACLs and uploading public objects *(Recommended)*
□ Remove public access granted through public ACLs *(Recommended)*

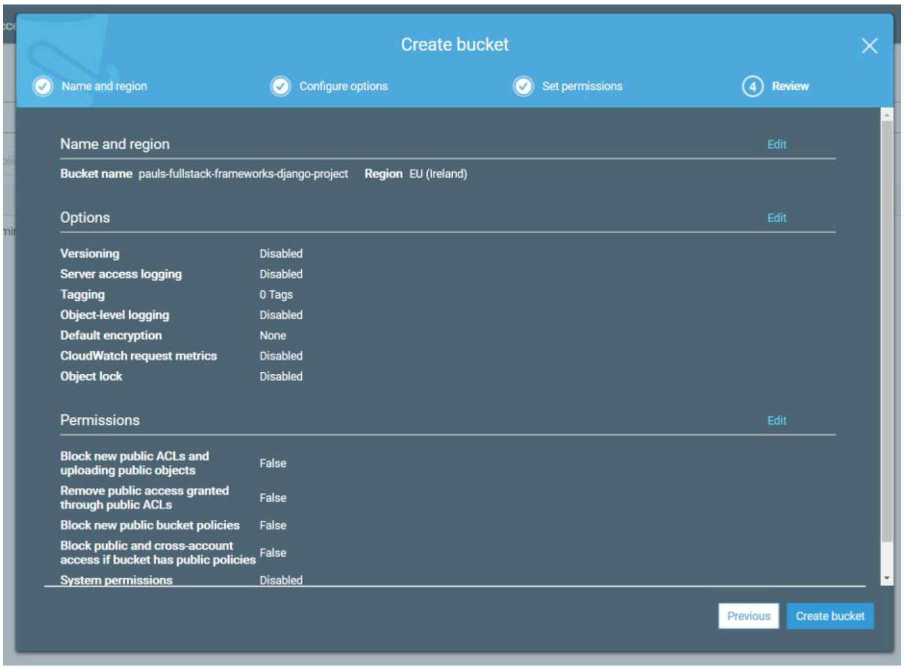
Manage public bucket policies for this bucket

□ Block new public bucket policies *(Recommended)*
□ Block public and cross-account access if bucket has public policies *(Recommended)*

Manage system permissions

Do not grant Amazon S3 Log Delivery group write access to this bucket

- Because we actually want this to be accessed by the public, we need to untick all of these four checkboxes and then click next button.



Name and region

Bucket name pauls-fullstack-frameworks-django-project **Region** EU (Ireland)

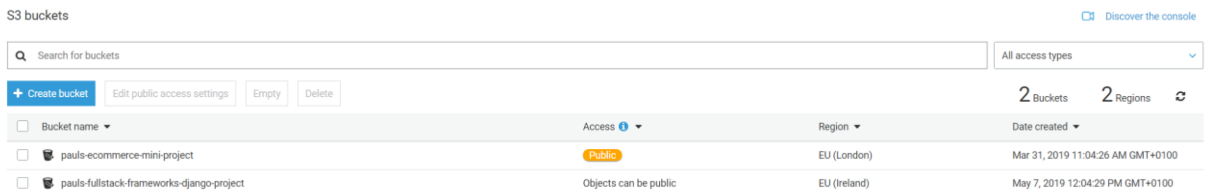
Options

Versioning	Disabled
Server access logging	Disabled
Tagging	0 Tags
Object-level logging	Disabled
Default encryption	None
CloudWatch request metrics	Disabled
Object lock	Disabled

Permissions

Block new public ACLs and uploading public objects	False
Remove public access granted through public ACLs	False
Block new public bucket policies	False
Block public and cross-account access if bucket has public policies	False
System permissions	Disabled

- Then click on Create bucket button

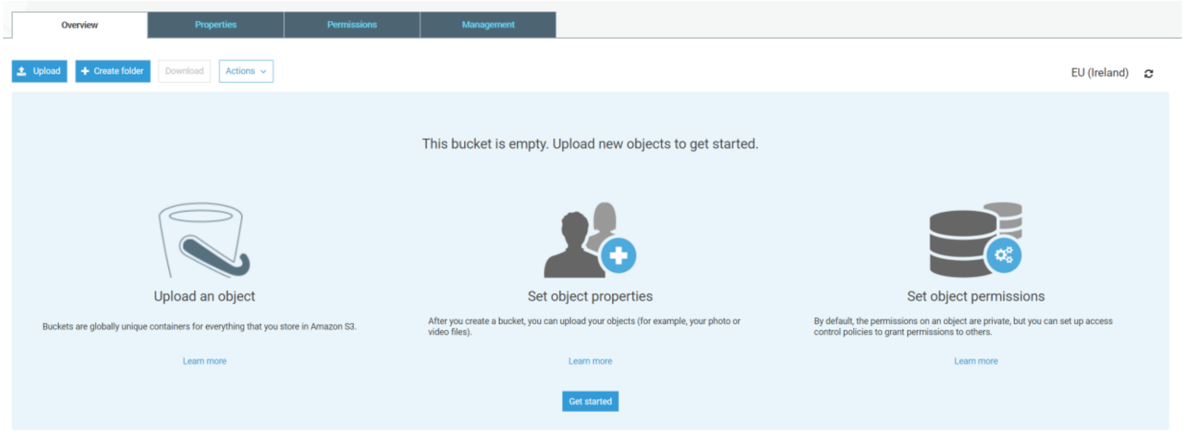


S3 buckets

2 Buckets 2 Regions

□ Bucket name	Access	Region	Date created
□ pauls-ecommerce-mini-project	Public	EU (London)	Mar 31, 2019 11:04:26 AM GMT+0100
□ pauls-fullstack-frameworks-django-project	Objects can be public	EU (Ireland)	May 7, 2019 12:04:29 PM GMT+0100

- Click on the new bucket name.



This bucket is empty. Upload new objects to get started.

Upload an object

Buckets are globally unique containers for everything that you store in Amazon S3.

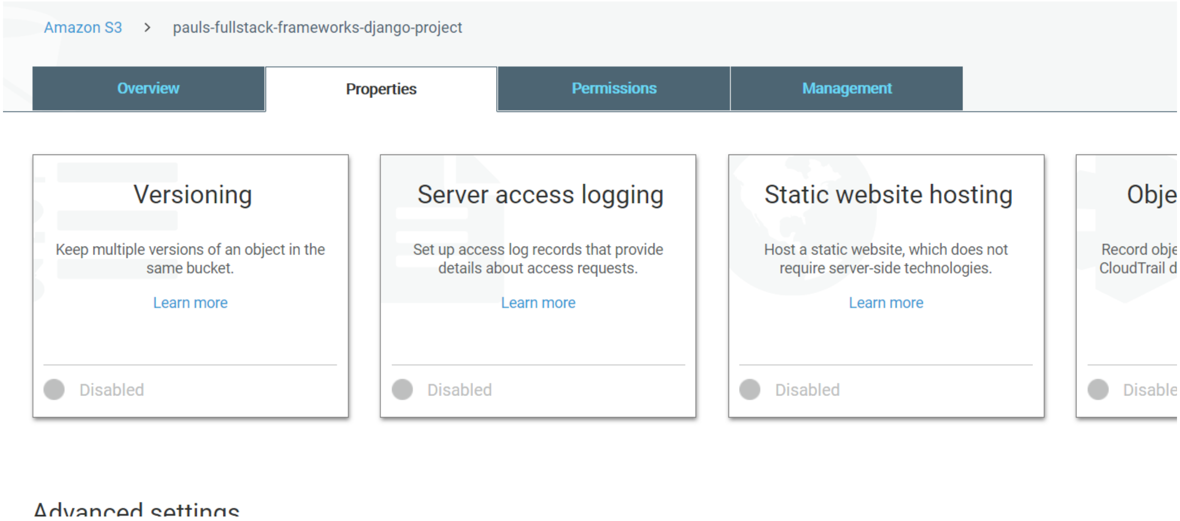
Set object properties

After you create a bucket, you can upload your objects (for example, your photo or video files).

Set object permissions

By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others.

- Click on Properties



Versioning

Keep multiple versions of an object in the same bucket.

Disabled

Server access logging

Set up access log records that provide details about access requests.

Disabled

Static website hosting

Host a static website, which does not require server-side technologies.

Disabled

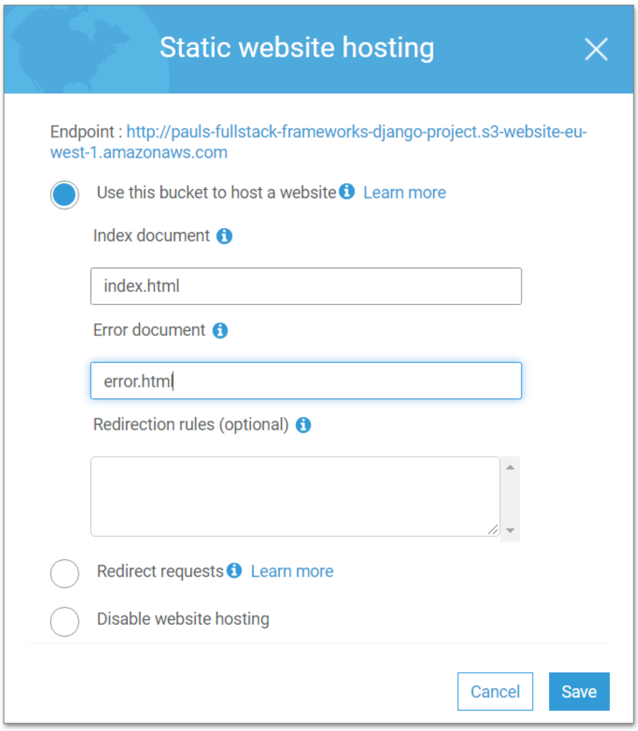
Obje

Record obje
CloudTrail d

Disable

Advanced settings

- Click on Static website hosting



Static website hosting

Endpoint : http://pauls-fullstack-frameworks-django-project.s3-website-eu-west-1.amazonaws.com

● Use this bucket to host a website

Index document

index.html

Error document

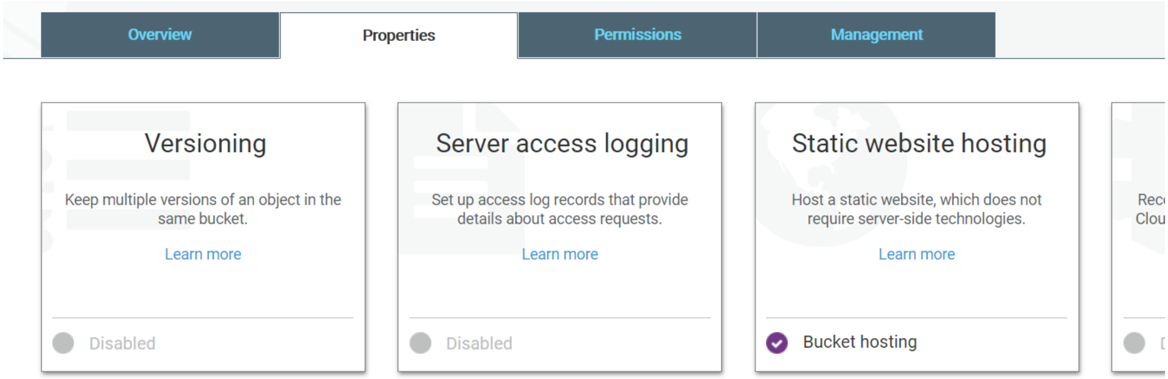
error.html

Redirection rules (optional)

○ Redirect requests

○ Disable website hosting

- Click on Use this bucket to host a website , Enter index.html in index document box and error.html in Error document then click on Save button.



Versioning

Keep multiple versions of an object in the same bucket.

Learn more

○ Disabled

Server access logging

Set up access log records that provide details about access requests.

Learn more

○ Disabled

Static website hosting

Host a static website, which does not require server-side technologies.

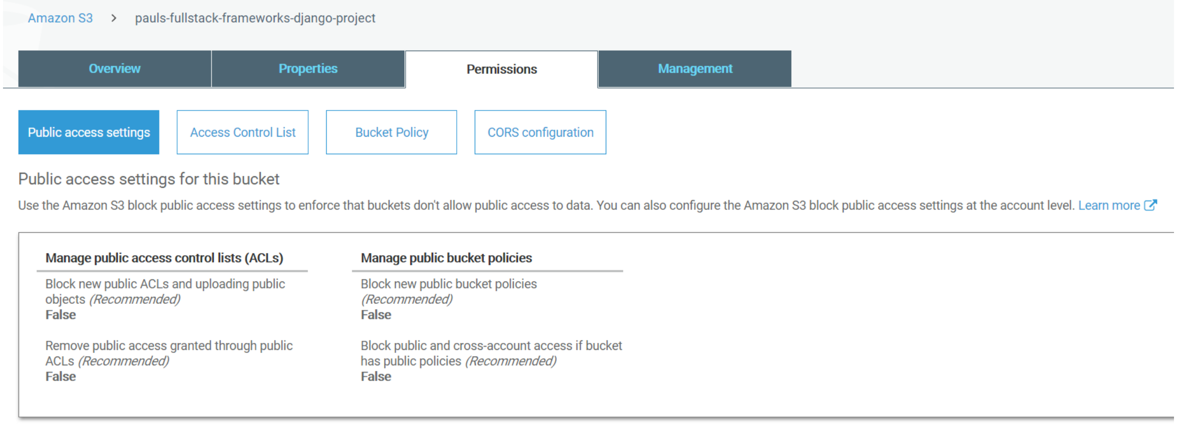
Learn more

✓ Bucket hosting

Rec
Clou

○ [

- Click on Permissions tab

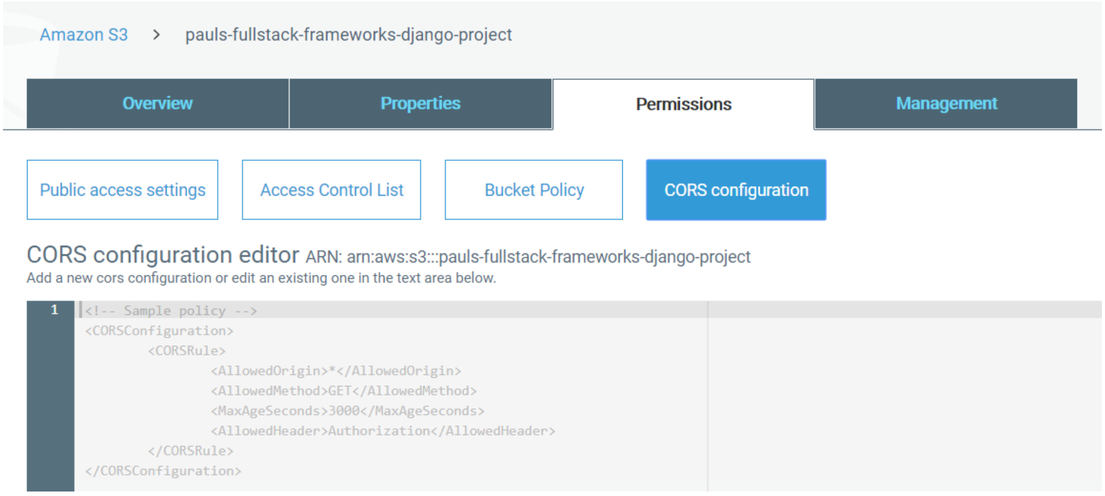


Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more

Manage public access control lists (ACLs)	Manage public bucket policies
Block new public ACLs and uploading public objects *(Recommended)* False	Block new public bucket policies *(Recommended)* False
Remove public access granted through public ACLs *(Recommended)* False	Block public and cross-account access if bucket has public policies *(Recommended)* False

- Click on CORS configuration button

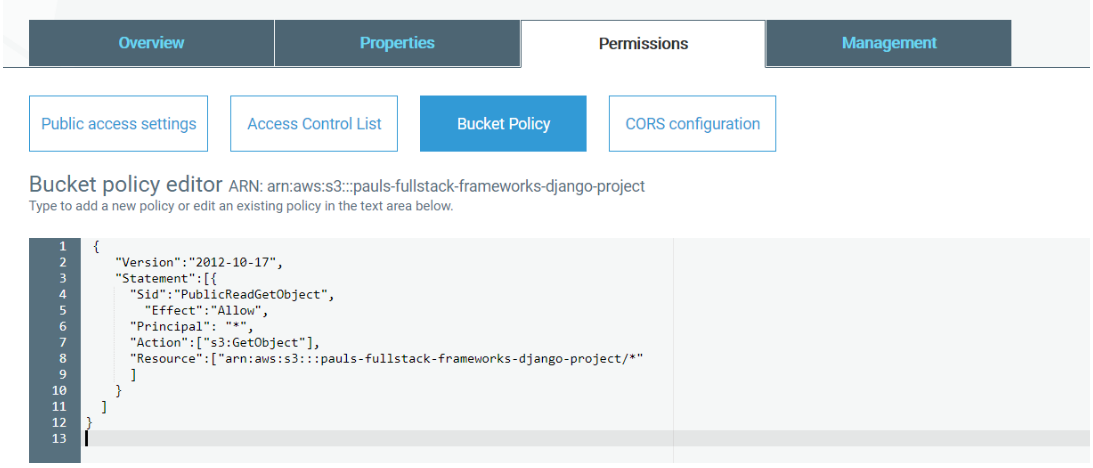


CORS configuration editor ARN: arn:aws:s3:::pauls-fullstack-frameworks-django-project

Add a new cors configuration or edit an existing one in the text area below.

```
<!-- Sample policy -->
<CORSConfiguration>
        <CORSRule>
                <AllowedOrigin>*</AllowedOrigin>
                <AllowedMethod>GET</AllowedMethod>
                <MaxAgeSeconds>3000</MaxAgeSeconds>
                <AllowedHeader>Authorization</AllowedHeader>
        </CORSRule>
</CORSConfiguration>
```

- Cut and paste the Default example below the video and then click on Save button

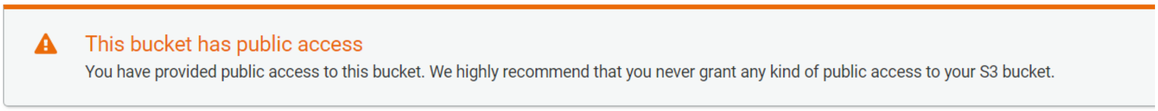


Bucket policy editor ARN: arn:aws:s3:::pauls-fullstack-frameworks-django-project

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

```
{
   "Version":"2012-10-17",
   "Statement":[{
     "Sid":"PublicReadGetObject",
       "Effect":"Allow",
     "Principal": "*",
     "Action":["s3:GetObject"],
     "Resource":["arn:aws:s3:::pauls-fullstack-frameworks-django-project/*"
     ]
   }
  ]
}
```

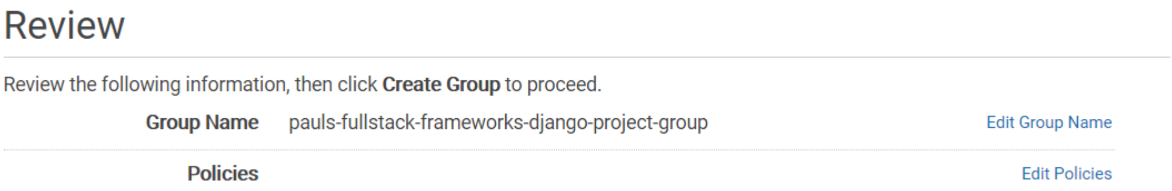
- Click on Bucket Policy
- Cut and paste the default example below the video
- Cut and paste the Buck policy editor into the Resource line no7 and keep the /* at the end of the line.
- Click on the Save button



This bucket has public access

You have provided public access to this bucket. We highly recommend that you never grant any kind of public access to your S3 bucket.

- Then you get a waring to say bucket has public access, but that's what we want.
- Click on Services on nav bar
- Type IAM in search box and press enter
- Click on Groups
- Click on Create New Group button
- Enter group name `pauls-fullstacks-frameworks-django-project-group` best to enter something that matches your bucket name.
- Click on Next Step button
- Click on Next Step button in attached policy page

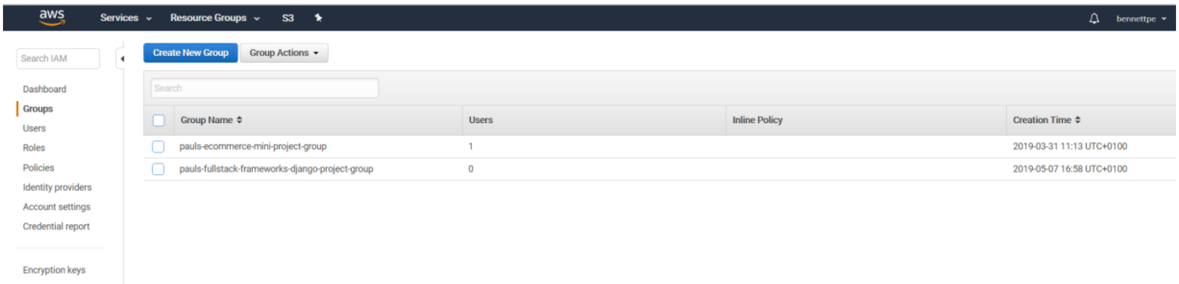


Review

Review the following information, then click **Create Group** to proceed.

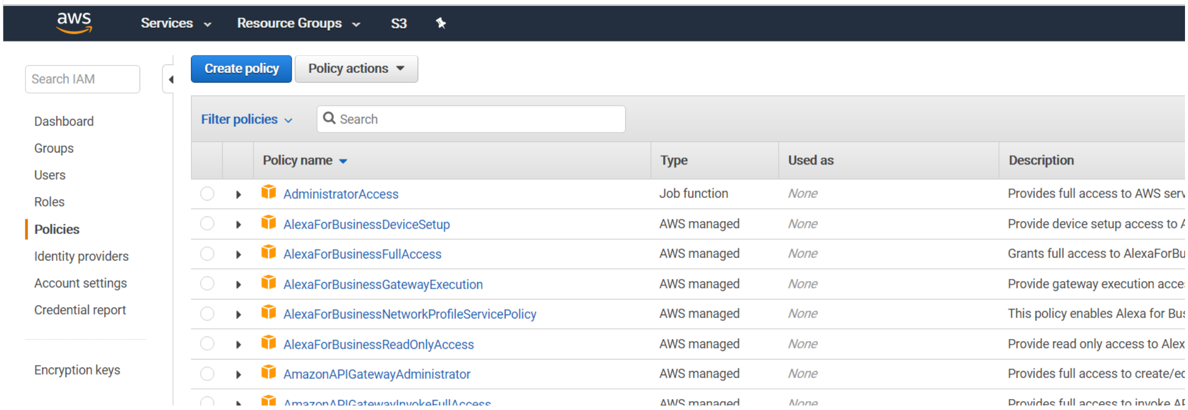
Group Name	pauls-fullstack-frameworks-django-project-group	Edit Group Name
Policies		Edit Policies

- Click on Create Group button



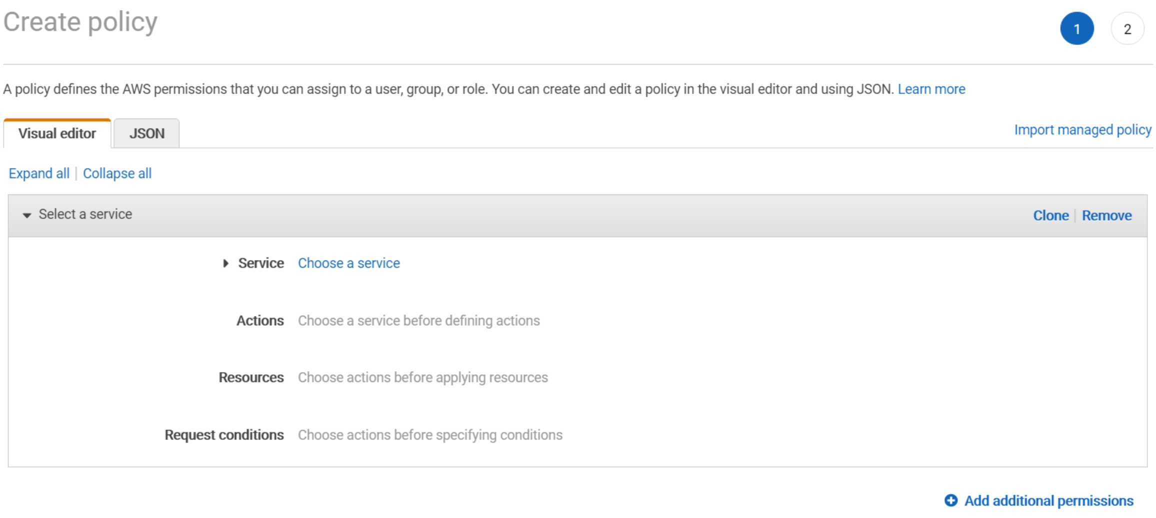
	Group Name	Users	Inline Policy	Creation Time
□	pauls-ecommerce-mini-project-group	1		2019-03-31 11:13 UTC+0100
□	pauls-fullstack-frameworks-django-project-group	0		2019-05-07 16:58 UTC+0100

- Click on Policies



		Policy name	Type	Used as	Description
○	▸	AdministratorAccess	Job function	None	Provides full access to AWS serv
○	▸	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to A
○	▸	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBu
○	▸	AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution acce
○	▸	AlexaForBusinessNetworkProfileServicePolicy	AWS managed	None	This policy enables Alexa for Bus
○	▸	AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to Alex
○	▸	AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/ec
○	▸	AmazonAPIGatewayInvokeFullAccess	AWS managed	None	Provides full access to invoke AP

- Click on Create policy button



Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

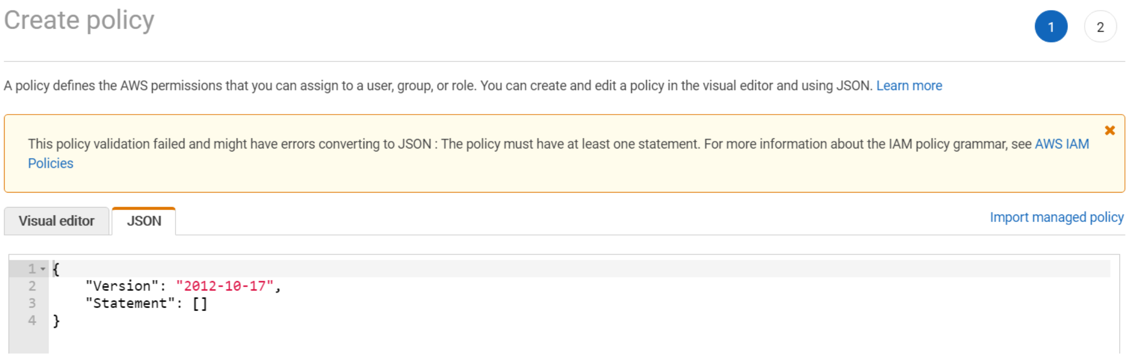
Service: Choose a service

Actions: Choose a service before defining actions

Resources: Choose actions before applying resources

Request conditions: Choose actions before specifying conditions

- Click on JSON tab



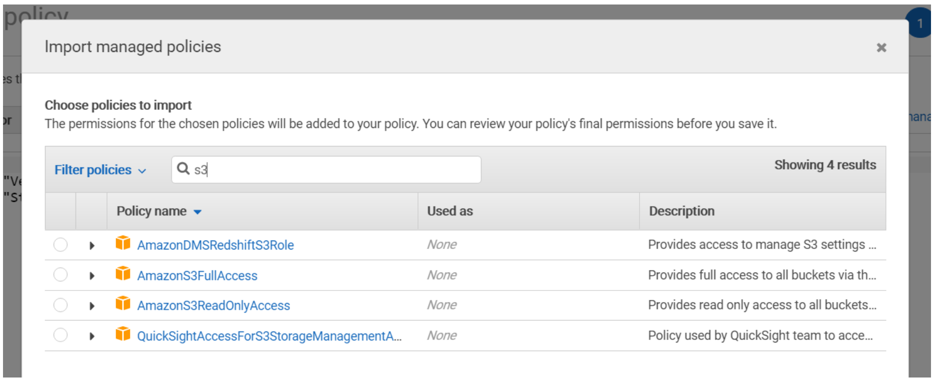
Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

This policy validation failed and might have errors converting to JSON : The policy must have at least one statement. For more information about the IAM policy grammar, see AWS IAM Policies

```
{
    "Version": "2012-10-17",
    "Statement": []
}
```

- Click on Import managed policy
-



Import managed policies

Choose policies to import

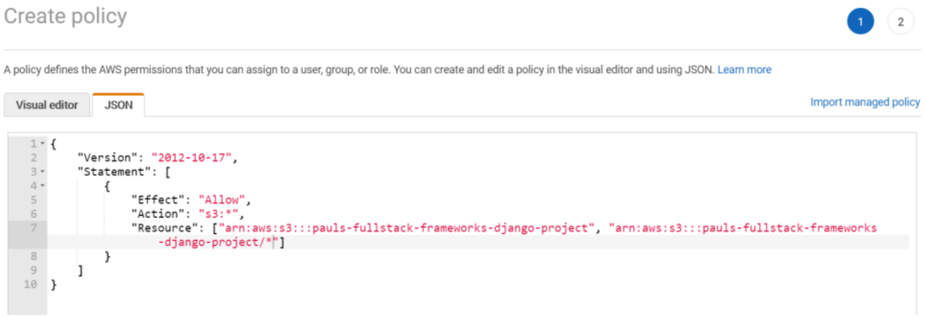
The permissions for the chosen policies will be added to your policy. You can review your policy's final permissions before you save it.

s3

Showing 4 results

		Policy name	Used as	Description
○	▸	AmazonDMSRedshiftS3Role	None	Provides access to manage S3 settings ...
○	▸	AmazonS3FullAccess	None	Provides full access to all buckets via th...
○	▸	AmazonS3ReadOnlyAccess	None	Provides read only access to all buckets...
○	▸	QuickSightAccessForS3StorageManagementA...	None	Policy used by QuickSight team to acce...

- Enter S3 in filter policies box
- Select AmazonS3FullAccess and click on import button

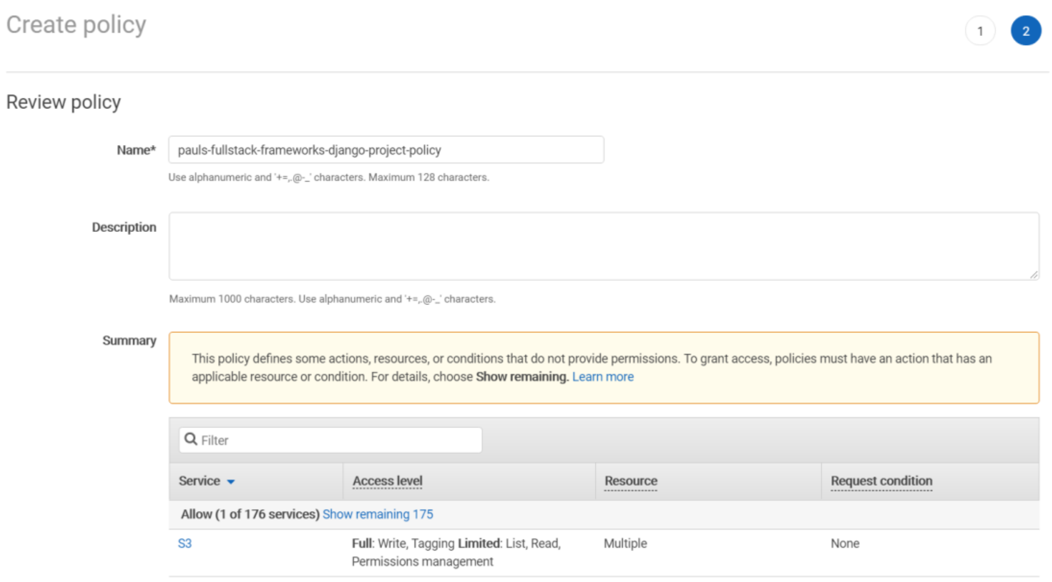


Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "s3:*",
7              "Resource": ["arn:aws:s3:::pauls-fullstack-frameworks-django-project", "arn:aws:s3:::pauls-fullstack-frameworks
                   -django-project/*"]
8          }
9      ]
10 }
```

- Update the resources line no 7 to include arn as above enclosed in quotes
- Click on Review policy button



Create policy

Review policy

Name* pauls-fullstack-frameworks-django-project-policy

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

Description

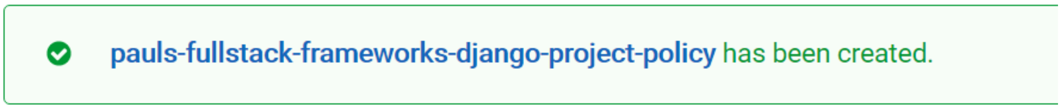
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining.** Learn more

Service	Access level	Resource	Request condition
Allow (1 of 176 services) Show remaining 175			
S3	**Full**: Write, Tagging **Limited**: List, Read, Permissions management	Multiple	None

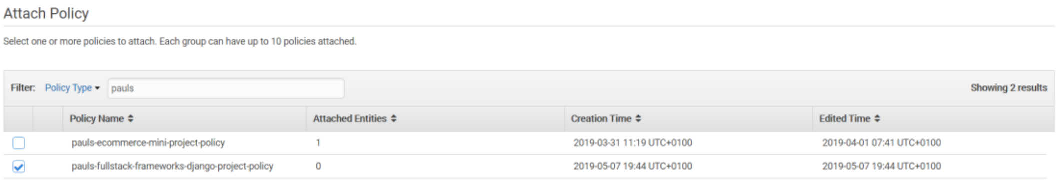
- Enter policy name in name box pauls-fullstack-frameworks-django-project-policy and click on Create policy button



pauls-fullstack-frameworks-django-project-policy has been created.

- Now that we have our policy, we need to add it to the group that we created earlier so click on group then click on group name

- Click on the Permissions tab, then click on Attach Policy button



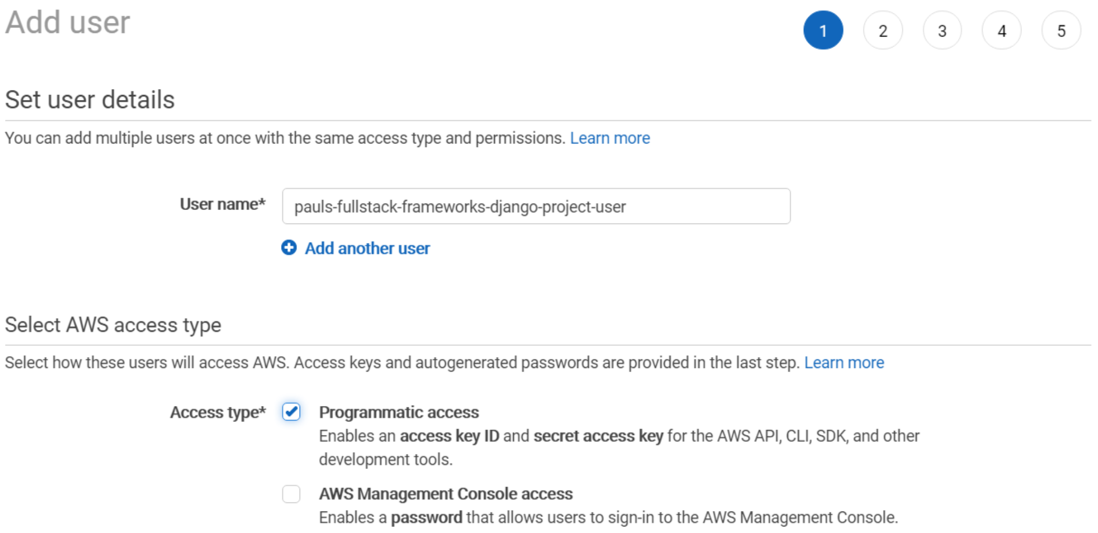
Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Filter: Policy Type ▾ pauls — Showing 2 results

	Policy Name	Attached Entities	Creation Time	Edited Time
□	pauls-ecommerce-mini-project-policy	1	2019-03-31 11:19 UTC+0100	2019-04-01 07:41 UTC+0100
☑	pauls-fullstack-frameworks-django-project-policy	0	2019-05-07 19:44 UTC+0100	2019-05-07 19:44 UTC+0100

- Search for policy in Policy Type box and then click checkbox then click on Attach Policy button
- So now we need to create a user so click on Users
- Click on Add user button



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* pauls-fullstack-frameworks-django-project-user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

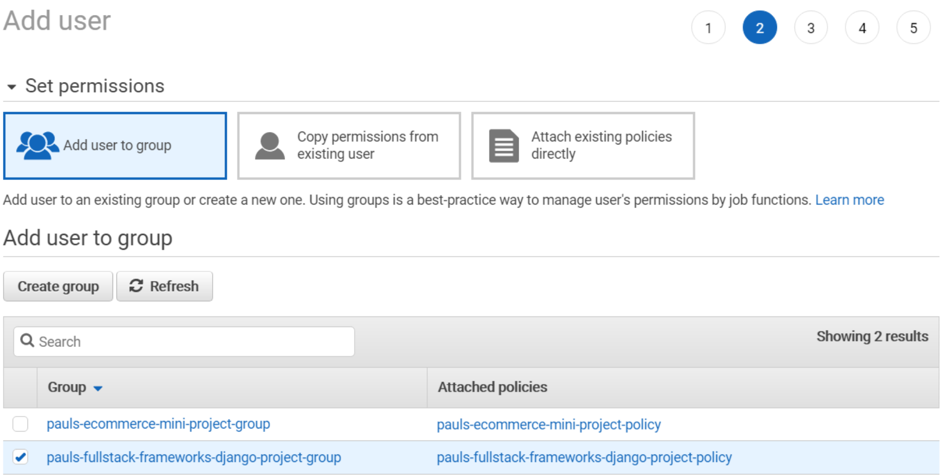
Access type* ☑ Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

- Add User name into box pauls-fullstack-frameworks-django-project-user click on Programmatic access checkbox and click on Next: Permissions button



Add user

Set permissions

Add user to group
Copy permissions from existing user
Attach existing policies directly

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

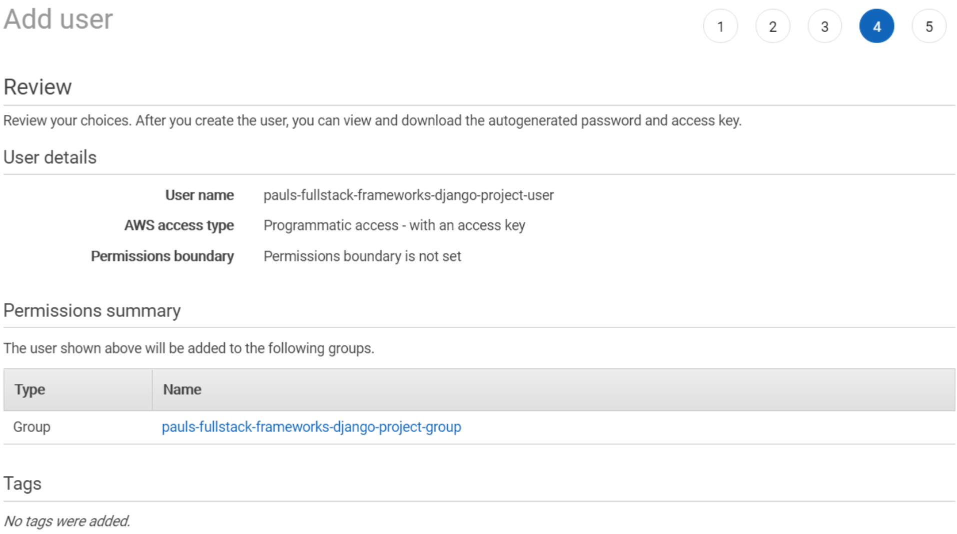
Add user to group

Create group
Refresh
Search

Showing 2 results

	Group	Attached policies
□	pauls-ecommerce-mini-project-group	pauls-ecommerce-mini-project-policy
☑	pauls-fullstack-frameworks-django-project-group	pauls-fullstack-frameworks-django-project-policy

- Click on the checkbox of the Group we just created and click on Next: Tags button
- We don't need any keys so click on Next: Review button



Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	pauls-fullstack-frameworks-django-project-user
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

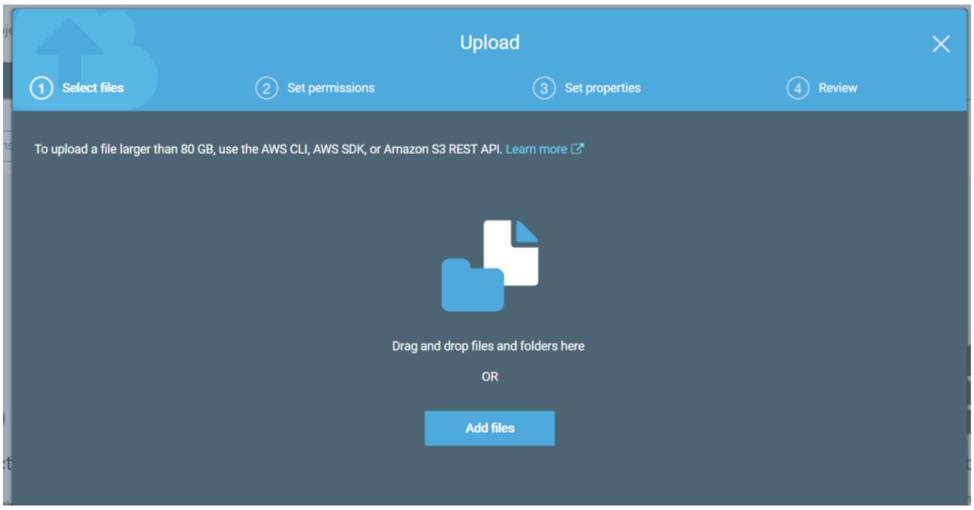
The user shown above will be added to the following groups.

Type	Name
Group	pauls-fullstack-frameworks-django-project-group

Tags

No tags were added.

- Click on Create user button
- Now we download the .csv file by clicking on Download .csv button as this will download the keys we need for our software.
  We need to keep this csv file safe because if we lose it, we would have to delete the user and create a new user with new access keys
  Now go back to S3
- We are going to test it by trying to Upload a file so click on Upload button



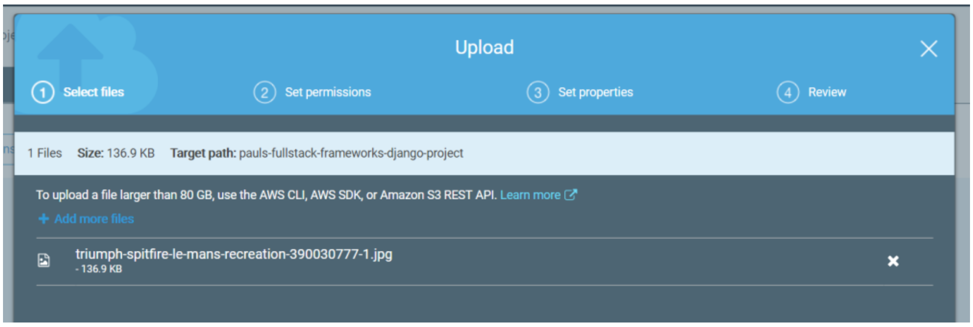
Upload

To upload a file larger than 80 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more

Drag and drop files and folders here

OR

- Click on Add files button and select a file

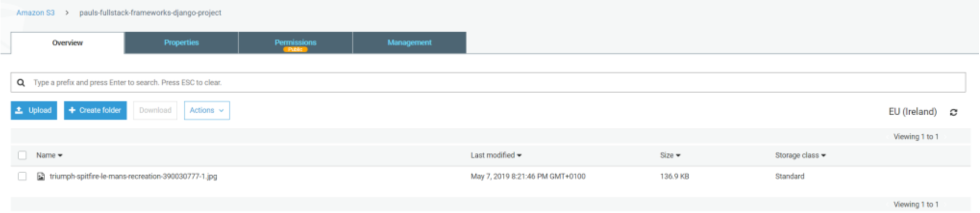


1 Files Size: 136.9 KB Target path: pauls-fullstack-frameworks-django-project

To upload a file larger than 80 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more

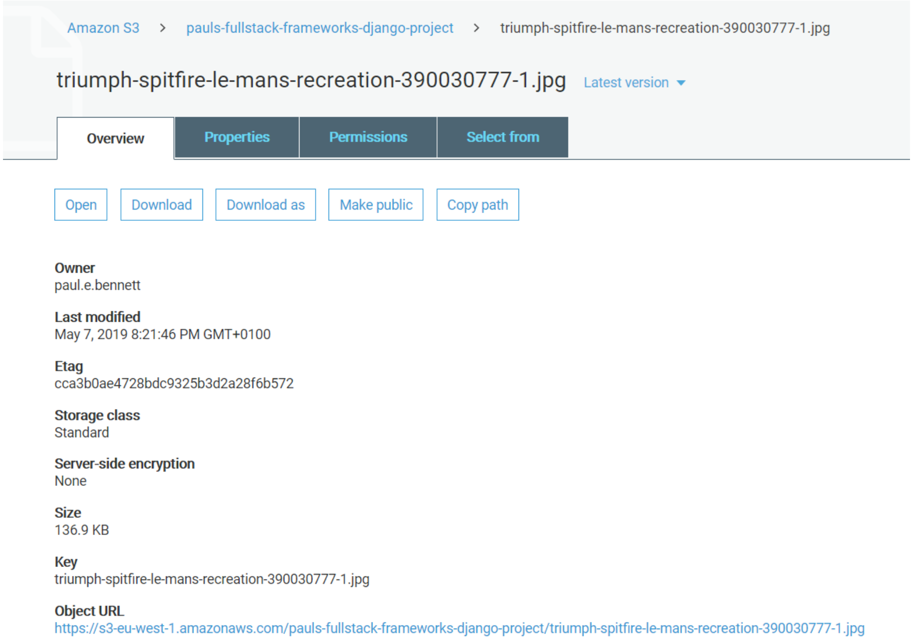
triumph-spitfire-le-mans-recreation-390030777-1.jpg
- 136.9 KB

- Then click on Upload button


Amazon S3 > pauls-fullstack-frameworks-django-project
Overview
Properties
Permissions
Public
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
EU (Ireland)
Viewing 1 to 1

Name	Last modified	Size	Storage class
triumph-spitfire-le-mans-recreation-390030777-1.jpg	May 7, 2019 8:21:46 PM GMT+0100	136.9 KB	Standard

- If we click on it, we can see we're provided with a URL and that URL will allow access to that image from anywhere on the web


Amazon S3 > pauls-fullstack-frameworks-django-project > triumph-spitfire-le-mans-recreation-390030777-1.jpg
Latest version
Overview
Properties
Permissions
Select from
Open
Download
Download as
Make public
Copy path

triumph-spitfire-le-mans-recreation-390030777-1.jpg

Owner
paul.e.bennett

Last modified
May 7, 2019 8:21:46 PM GMT+0100

Etag
cca3b0ae4728bdc9325b3d2a28f6b572

Storage class
Standard

Server-side encryption
None

Size
136.9 KB

Key
triumph-spitfire-le-mans-recreation-390030777-1.jpg

Object URL
https://s3-eu-west-1.amazonaws.com/pauls-fullstack-frameworks-django-project/triumph-spitfire-le-mans-recreation-390030777-1.jpg