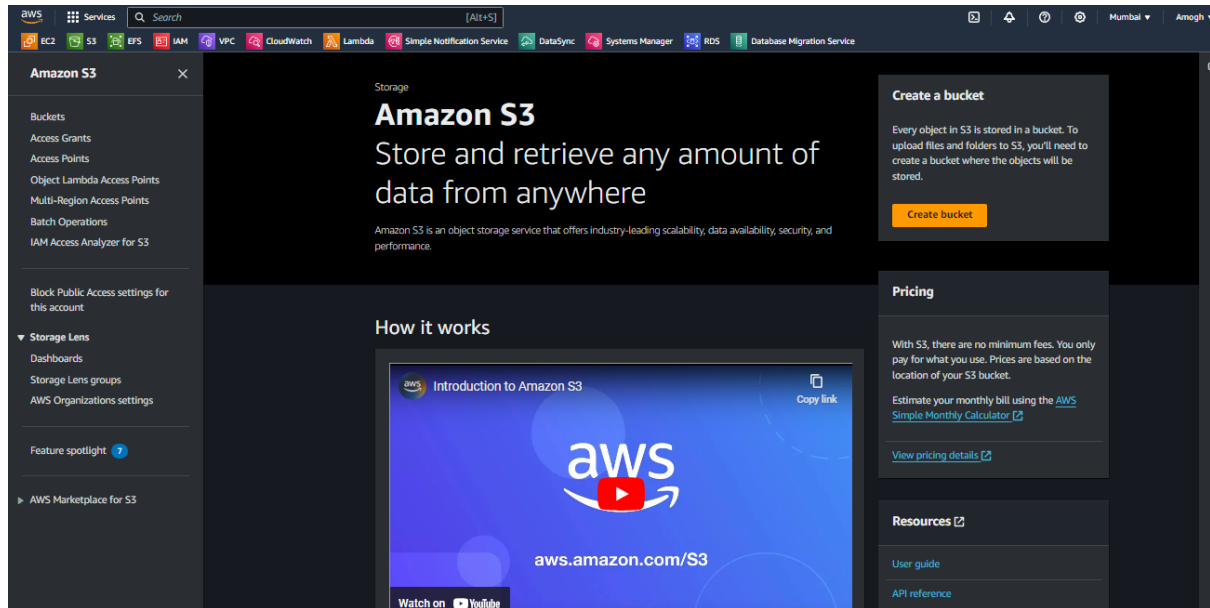


# TASK

Create 2 S3 buckets and 2 IAM Users Give policy to both the IAM user in which one of the IAM user have full access of s3 service and the other IAM user having only read-only access to bucket 2(Single bucket).

Step 1: Create 2 s3 buckets in AWS console using root user authentication



- Services

Search

[Alt+S]

EC2

S3

EFS

EMR

VPC

CloudWatch

Lambda

Simple Notification Service

DataSync

Systems Manager

RDS

Database Migration Service

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 name

info

mpowbucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: us-/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 the bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

### 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.

### 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.

Cancel

Create bucket

- Services

[Alt+S]

EC2

S3

EFS

EMR

IAM

VPC

CloudWatch

Lambda

Simple Notification Service

DataSync

Systems Manager

RDS

Database Migration Service

Mumbai

Amogh

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 name info

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

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 permission management and reduces costs.

Feedback

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.

If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more.

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 to 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

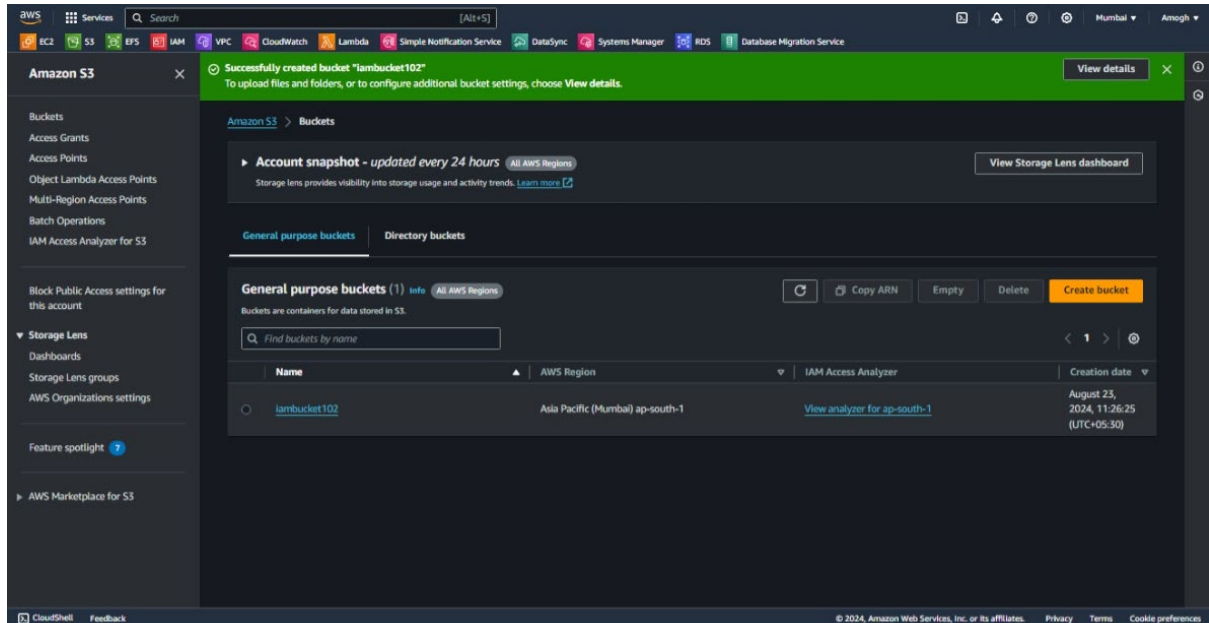
Advanced settings

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

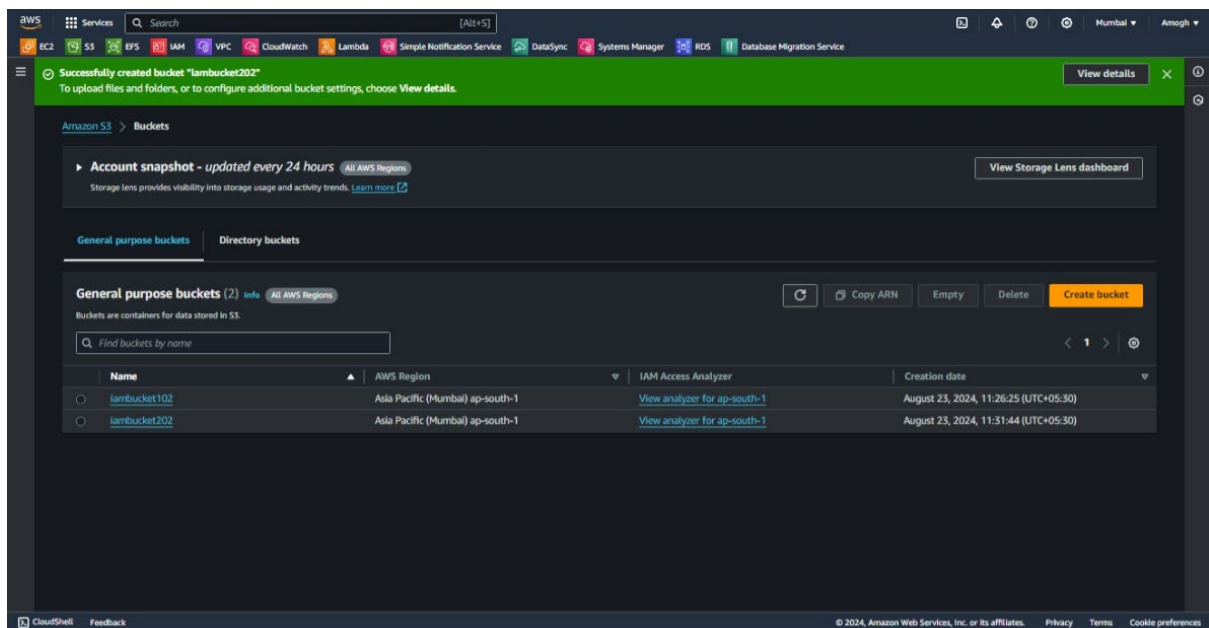
Cancel

Create bucket

- Now 1 s3 bucket is created

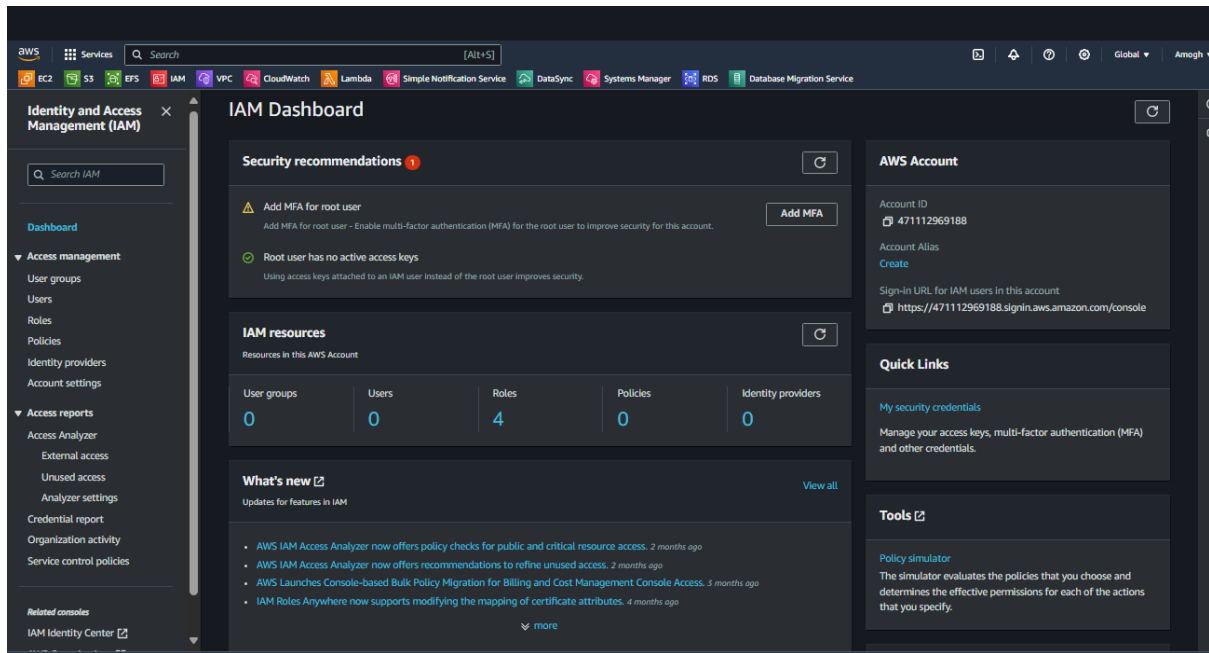


Similarly, we have to create 2<sup>nd</sup> bucket.

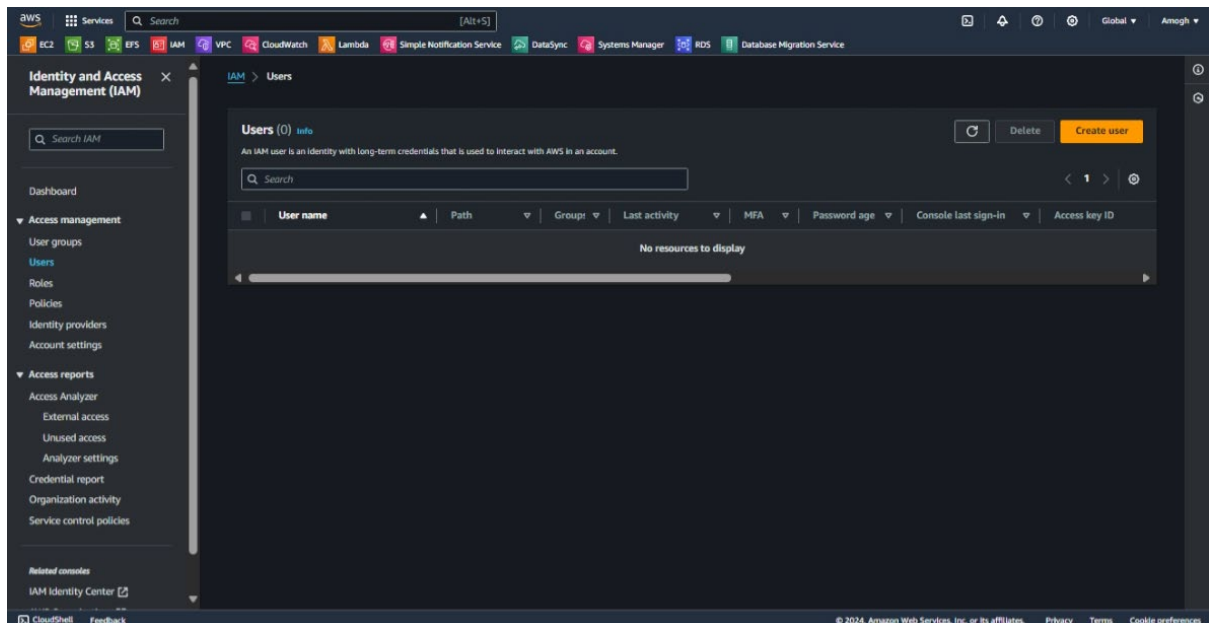


- We have successfully created the 2<sup>nd</sup> bucket as well.

Step 2: Now after successfully creating 2 buckets, we have to search “IAM” in the Service desk and click on the service.



- Here click on the option “Users” and choose the option and click on “Create user”.



- After going inside “**Create user**” page, name the username and click on the checkbox below to “**Provide user access to the AWS Management Console**” so that password can be created by us for the IAM user.

**Specify user details**

**User details**

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, =, @, \_ (hyphen)

☐ **Provide user access to the AWS Management Console - optional**  
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

**Information** If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

- We are disabling the “**User must create a new password at next sign-in**” option.

**Specify user details**

**User details**

User name

iam1

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, =, @, \_ (hyphen)

☒ **Provide user access to the AWS Management Console - optional**  
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

**Information** Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended  
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ **I want to create an IAM user**  
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☐ Autogenerated password  
You can view the password after you create the user.

☒ **Custom password**  
Enter a custom password for the user.

Must be at least 8 characters long  
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ( ! @ # \$ % ^ & \* ( ) \_ + - (hyphen) = [ ] { } )

☐ Show password

**Information** If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**



- We need to attach policy for an IAM user. Here we should give s3FullAccess policy for 1<sup>st</sup> IAM user.

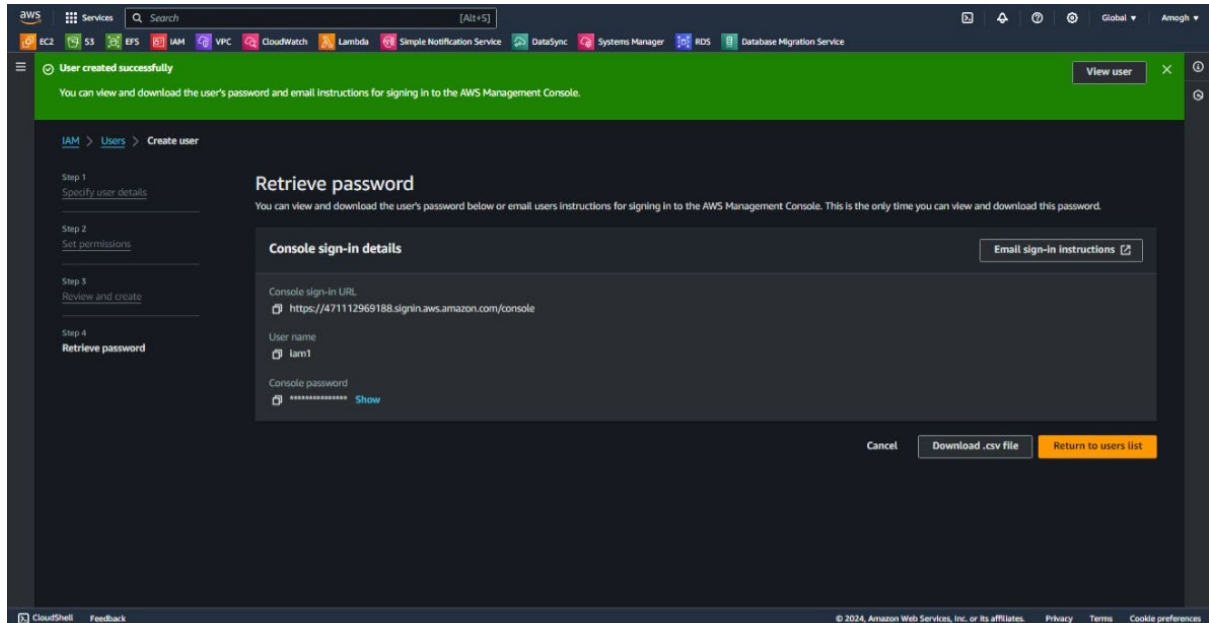
The screenshot shows the 'Set permissions' step in the AWS IAM console. The left sidebar indicates the current step is 'Set permissions'. The main content area has three options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected. Below this, a search for 's3' shows 12 matches. A table lists several AWS managed policies, with 'AmazonS3FullAccess' selected.

Policy name	Type	Attached entities
AmazonDMSRedshiftS3Role	AWS managed	0
<b>AmazonS3FullAccess</b>	AWS managed	0
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	0
AmazonS3OutpostsFullAccess	AWS managed	0
AmazonS3OutpostsReadOnlyAccess	AWS managed	0
AmazonS3ReadOnlyAccess	AWS managed	0

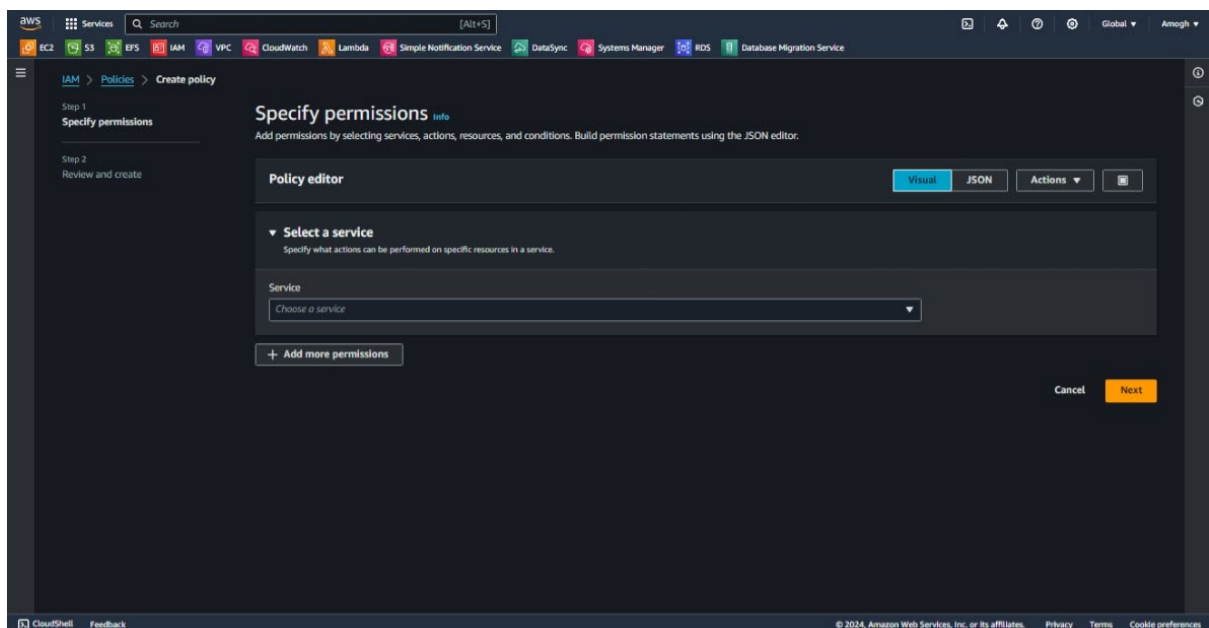
The screenshot shows the 'Review and create' step in the AWS IAM console. The left sidebar indicates the current step is 'Review and create'. The main content area shows the 'User details' section with 'User name' set to 'iam1', 'Console password type' set to 'Custom password', and 'Require password reset' set to 'No'. Below this is the 'Permissions summary' section, which shows 'AmazonS3FullAccess' as the attached policy. At the bottom, there is a 'Tags' section with an 'Add new tag' button. The 'Create user' button is highlighted in orange.

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

- 1<sup>st</sup> User is created. Now we should save the console sign-in details.

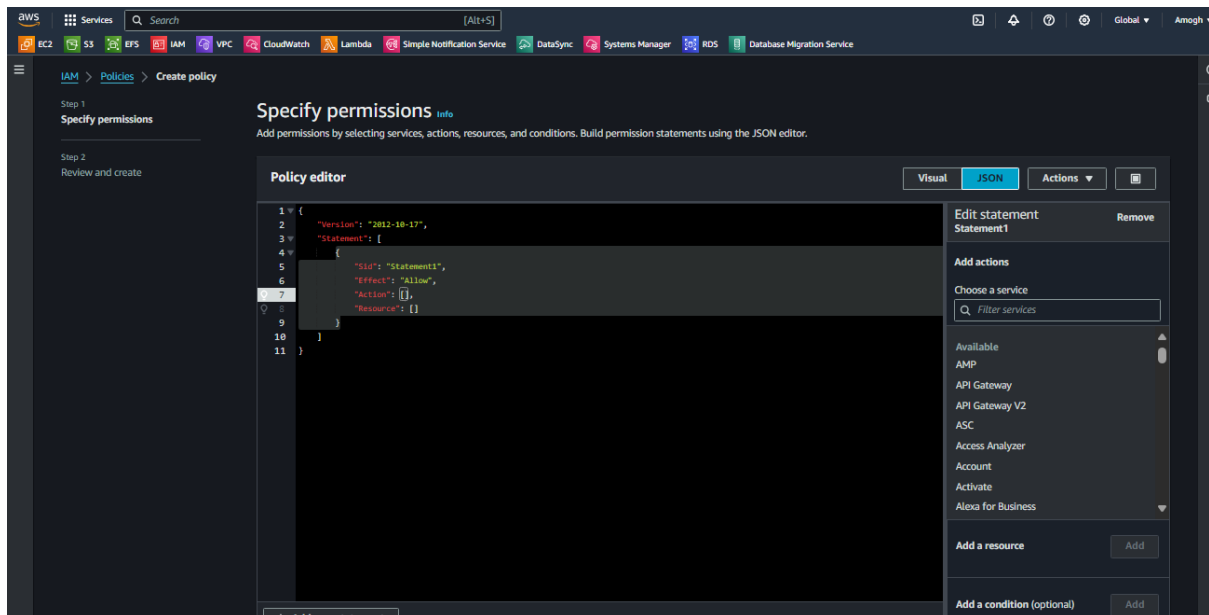


Step 3: After creating 1<sup>st</sup> IAM user with S3FullAccess policy, we should create a policy by clicking on “**Create policy**” option.

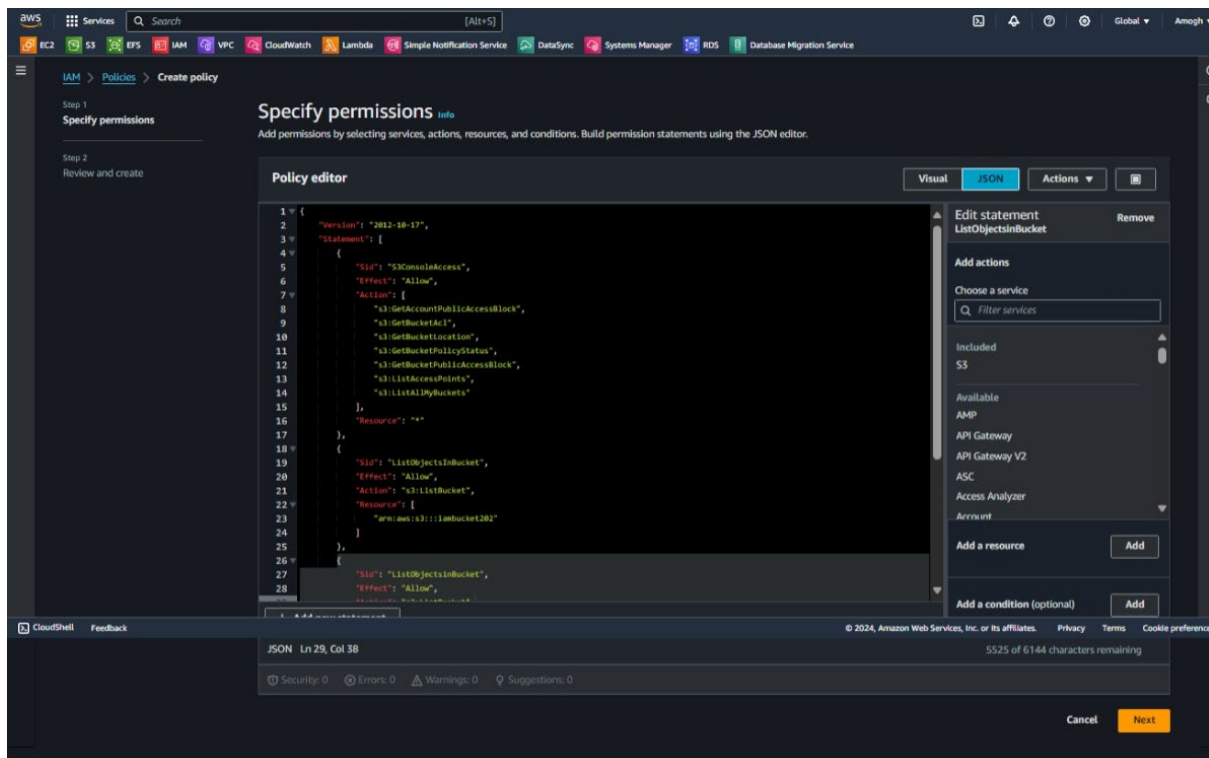




- Below is the default JSON policy



- Click on JSON to choose the JSON option and give the command for limited access to s3 bucket where the IAM user have only read-only Access to bucket 2



- The above policy code is mentioned below: (Note: Replace bucket names)

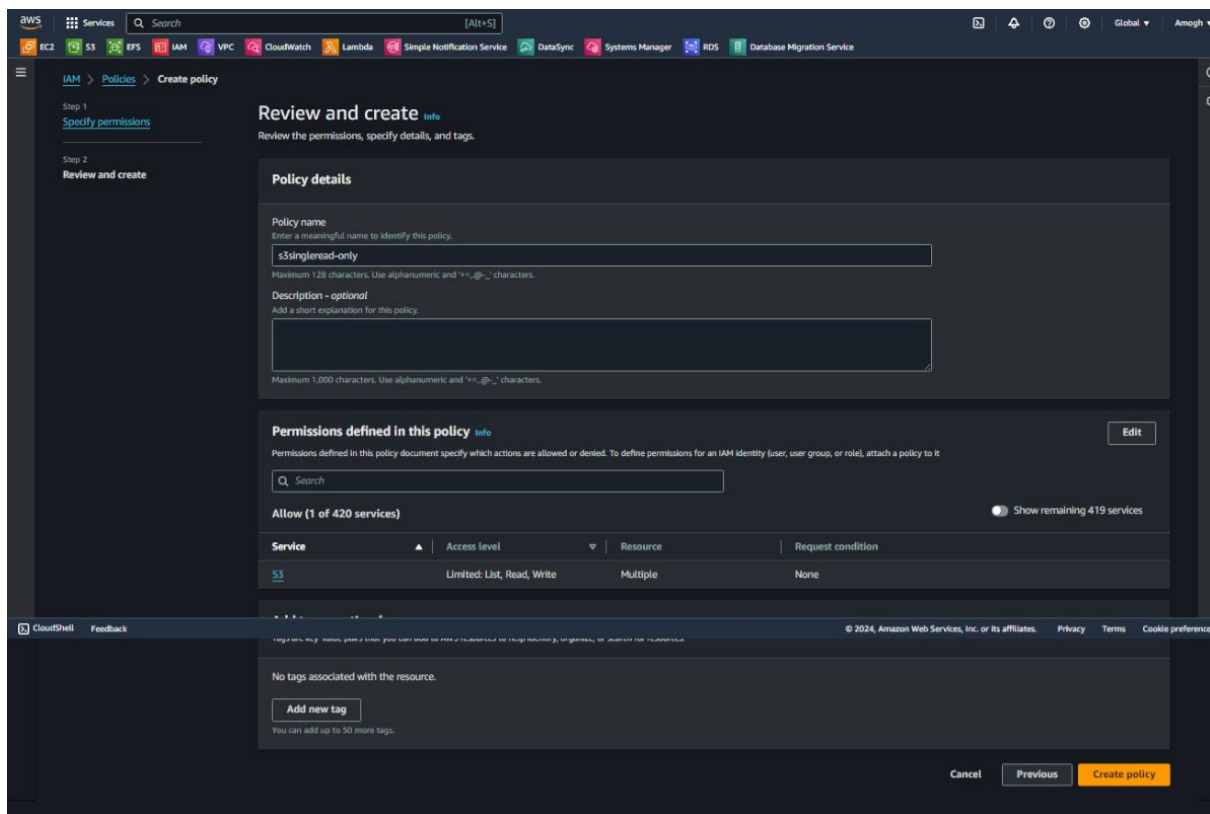
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ConsoleAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetAccountPublicAccessBlock",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:GetBucketPolicyStatus",
        "s3:GetBucketPublicAccessBlock",
        "s3:ListAccessPoints",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": [
        "arn:aws:s3:::iambucket202"
      ]
    },
    {
      "Sid": "ListObjectsinBucket",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
```

```

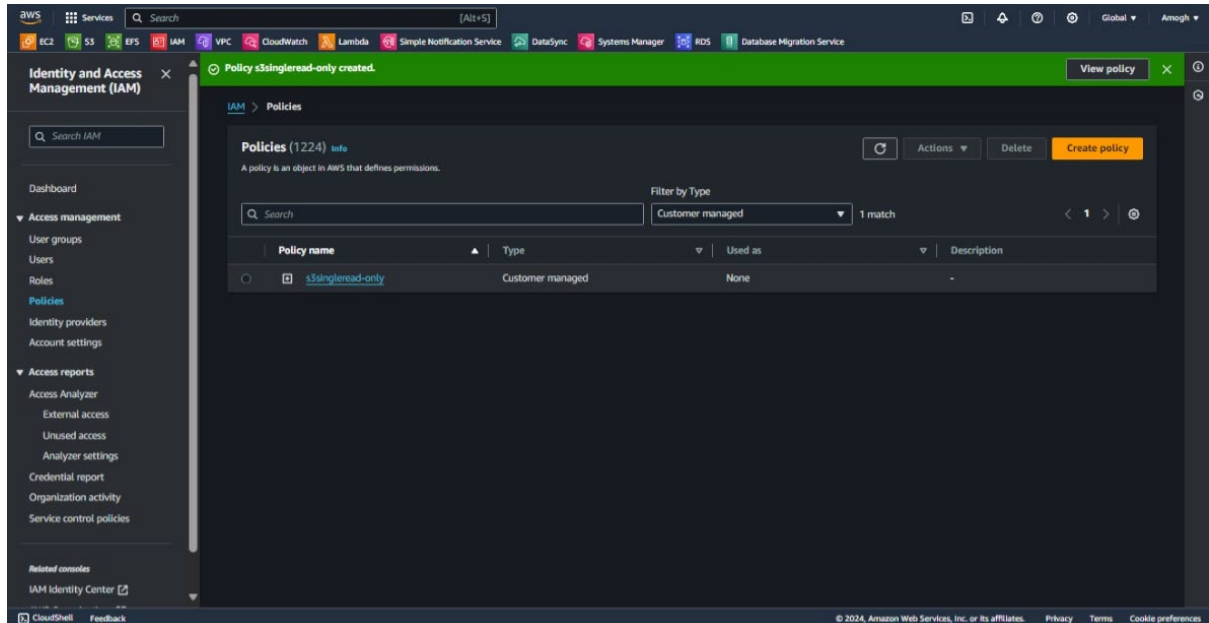
    "Resource": [
        "arn:aws:s3:::iambucket102"
    ]
},
{
    "Sid": "AllObjectActions",
    "Effect": "Allow",
    "Action": "s3:*Object",
    "Resource": [
        "arn:aws:s3:::iambucket102/*"
    ]
}
]
}

```

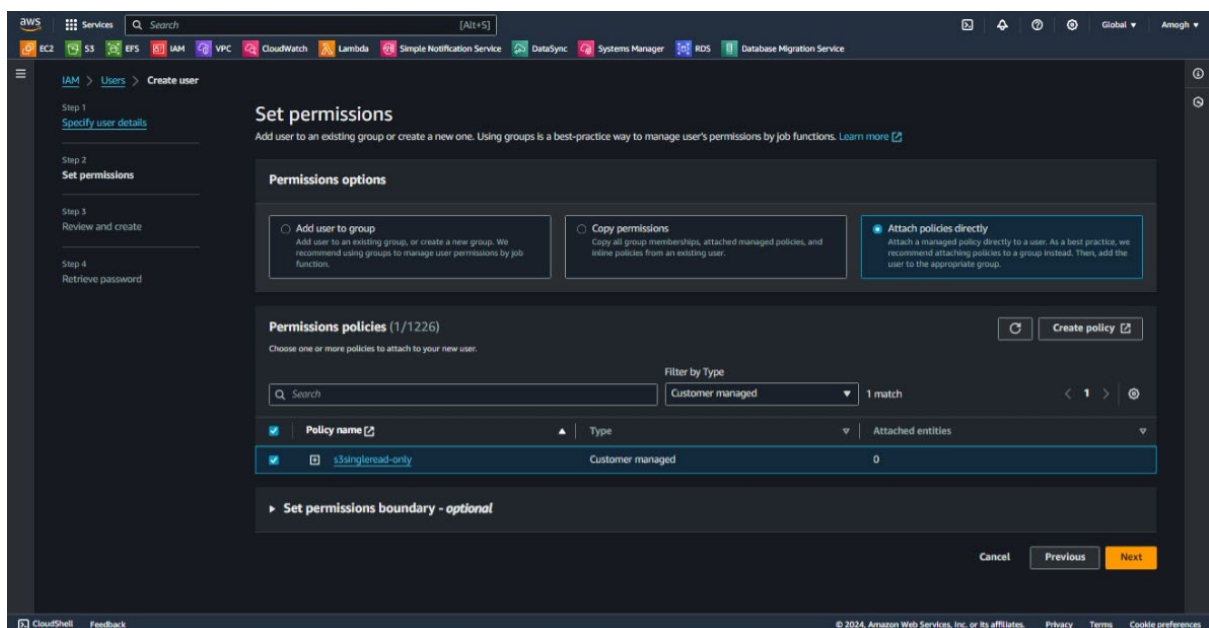
- Clicking Next takes us to naming the policy page

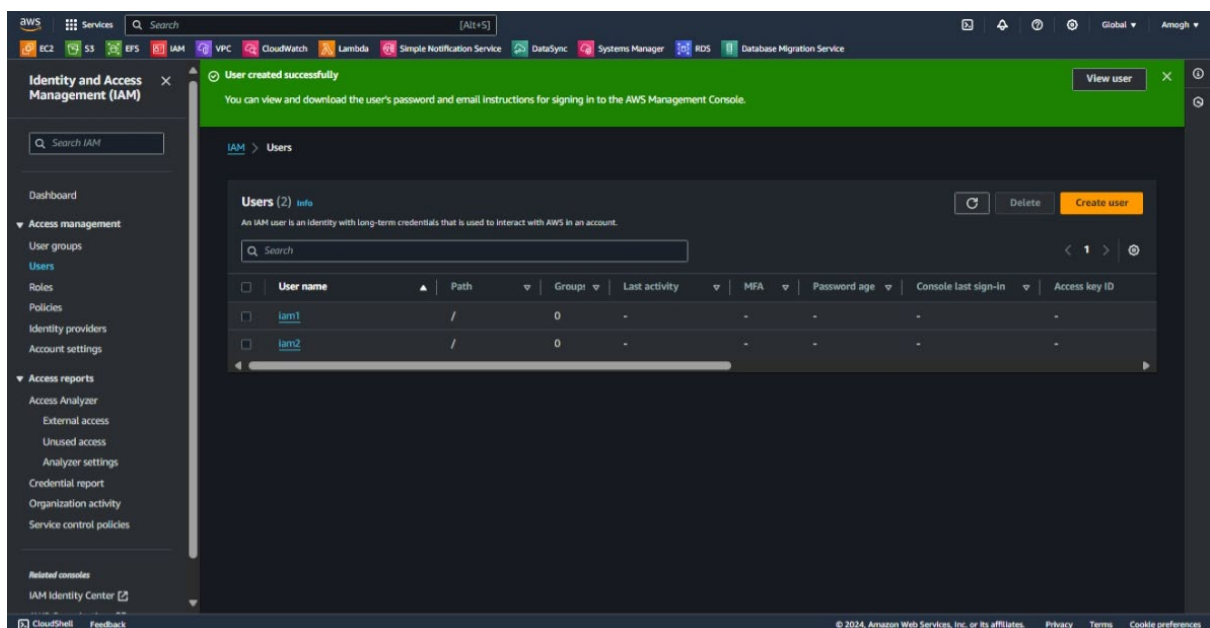
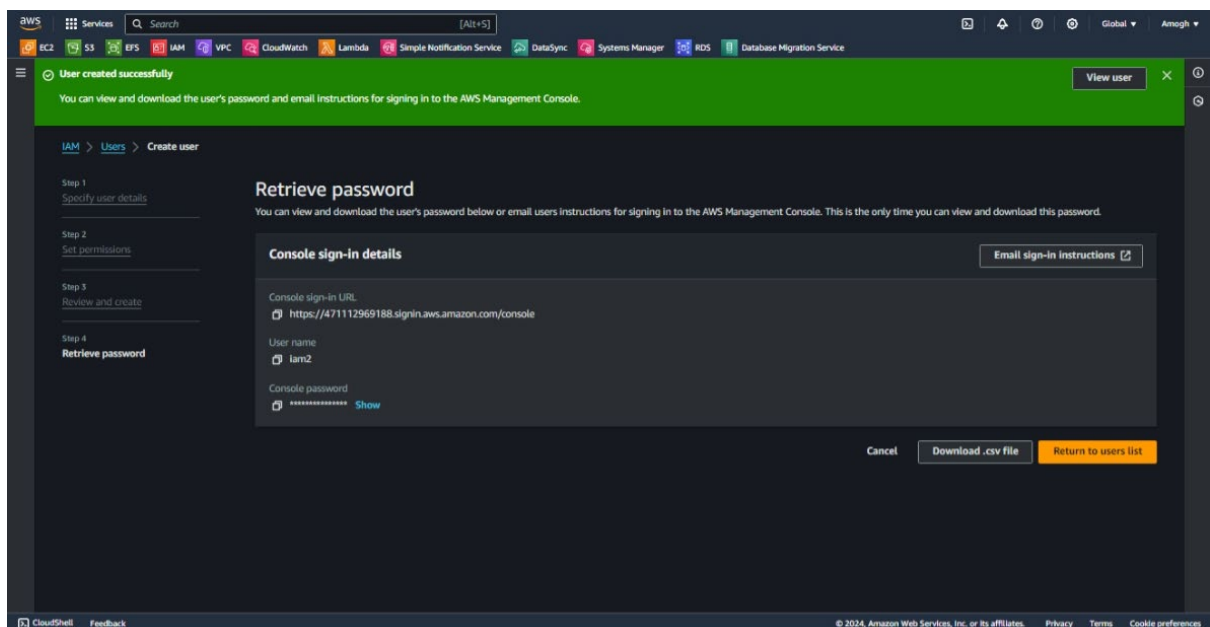
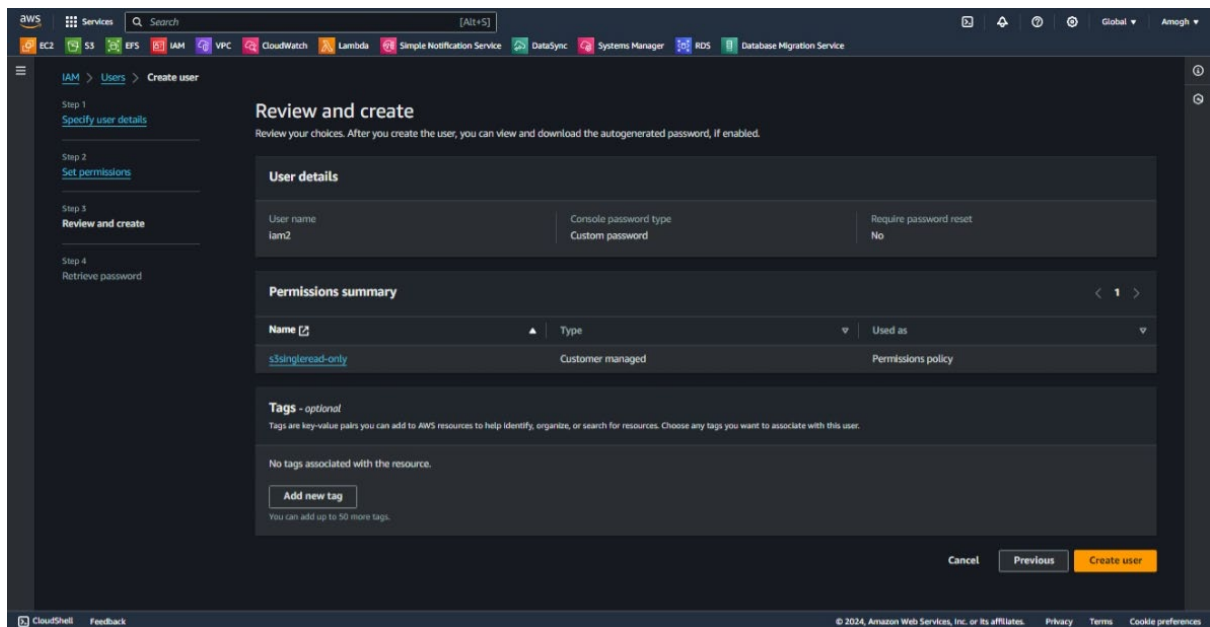


- Customer policy is created



- Now we need to create 2<sup>nd</sup> IAM user with the same procedure as we used in creating 1<sup>st</sup> IAM user but the only difference is in policy selection section. For 2<sup>nd</sup> IAM user, we should choose the customer managed policy that we have created with the name “s3singleread-only”. So, we should attach “s3singleread-only” to 2<sup>nd</sup> IAM user.

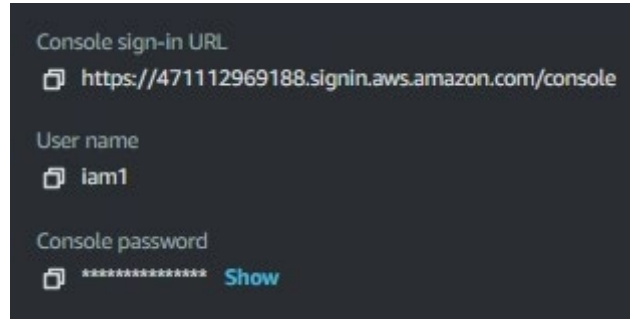




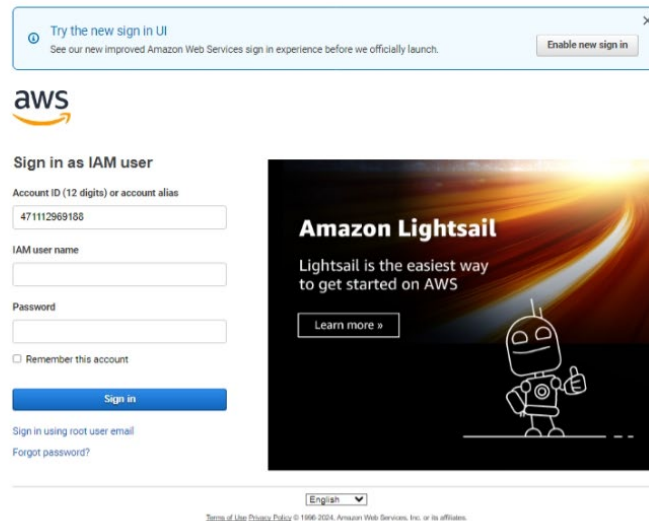
- After successfully creating 2 users and attaching the required policy and saving the credentials of IAM users' sign-in for both users, we should logout from root user and go to console login and sign-in to 1<sup>st</sup> IAM user

OR

Paste the link of the IAM user

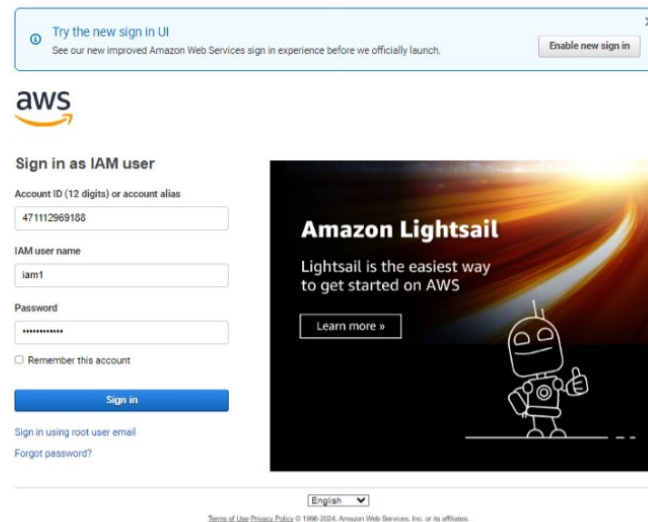


- Now, we should paste (or) type the username and password for signing-in. Here I have pasted the given link and signing in.



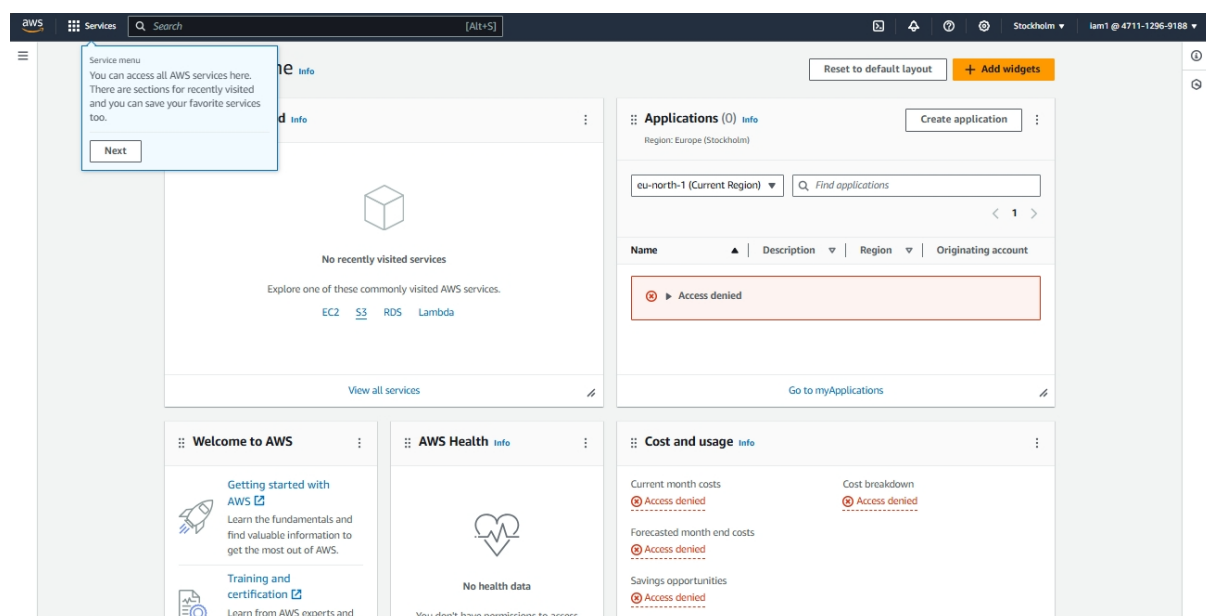


- Since, we pasted the link the “**Account ID**” will automatically get entered, if we to try sign-in directly from console we should type the Account ID manually.



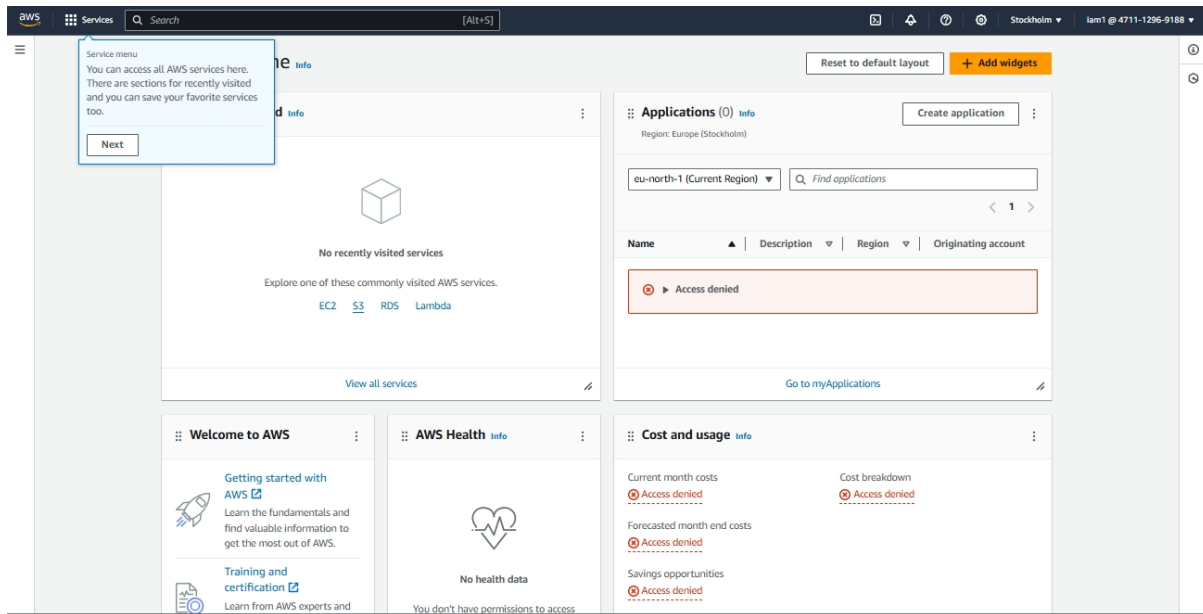
The image shows the AWS sign-in page for an IAM user. At the top, there is a notification banner about the new sign-in UI. Below it is the AWS logo. The main heading is "Sign in as IAM user". There are three input fields: "Account ID (12 digits) or account alias" with the value "471112969188", "IAM user name" with the value "iam1", and "Password" with masked characters. There is a checkbox for "Remember this account" and a "Sign in" button. Below the button are links for "Sign in using root user email" and "Forgot password?". On the right, there is a promotional banner for Amazon Lightsail. At the bottom, there is a language selector set to "English" and a link to the "Terms of Use Privacy Policy".

- After signing into the 1<sup>st</sup> IAM user account, this is the display page.

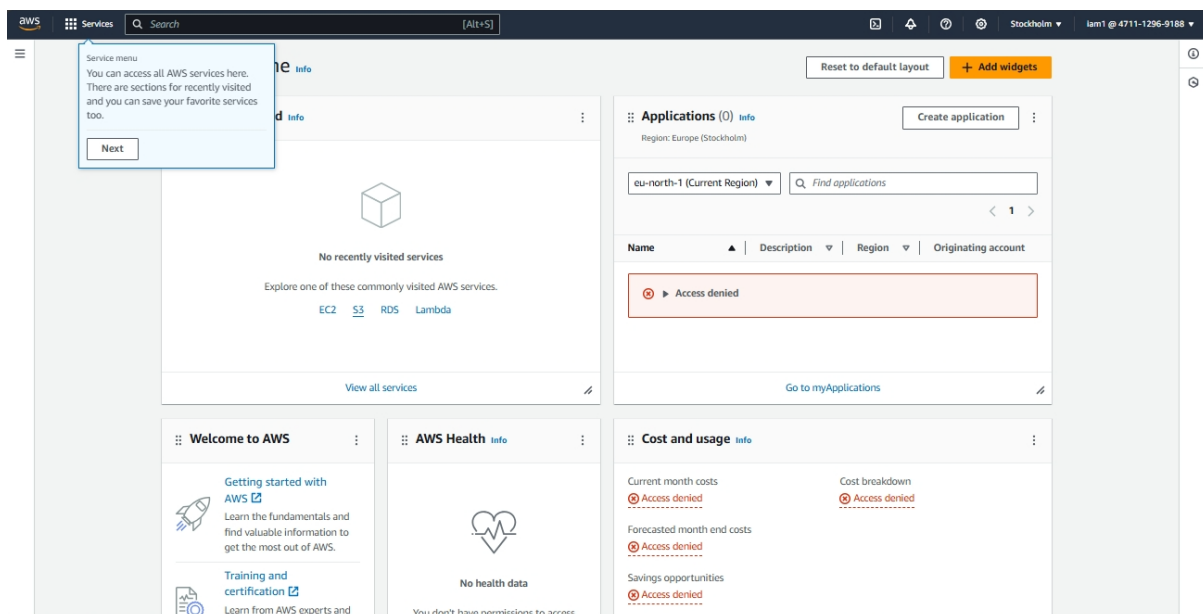


The image shows the AWS Management Console dashboard for the IAM user. The top navigation bar includes the AWS logo, a "Services" menu, a search bar, and the user's profile information. A "Service menu" tooltip is visible on the left. The main content area is divided into several sections: "No recently visited services" with links to EC2, S3, RDS, and Lambda; "Applications (0)" with a "Create application" button and a table showing "Access denied"; "Welcome to AWS" with links to "Getting started with AWS" and "Training and certification"; "AWS Health" showing "No health data"; and "Cost and usage" with sections for "Current month costs", "Forecasted month end costs", and "Savings opportunities", all showing "Access denied".

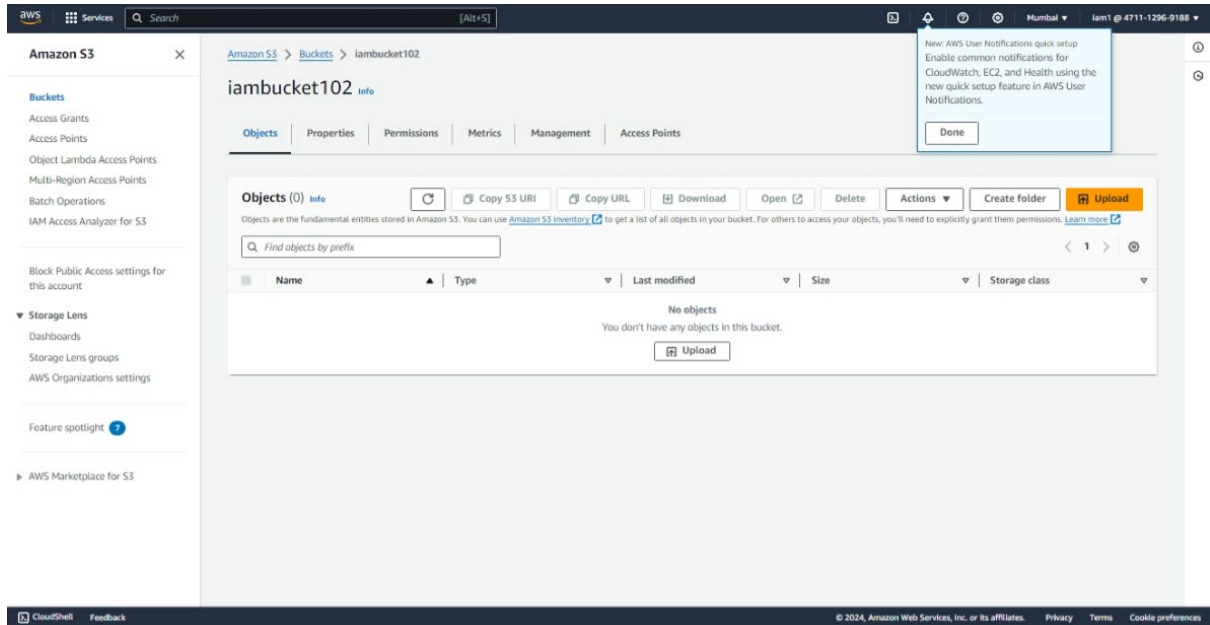
- It's better and safe to change the region to our default “Asia Pacific (Mumbai)ap-south-1” Region.



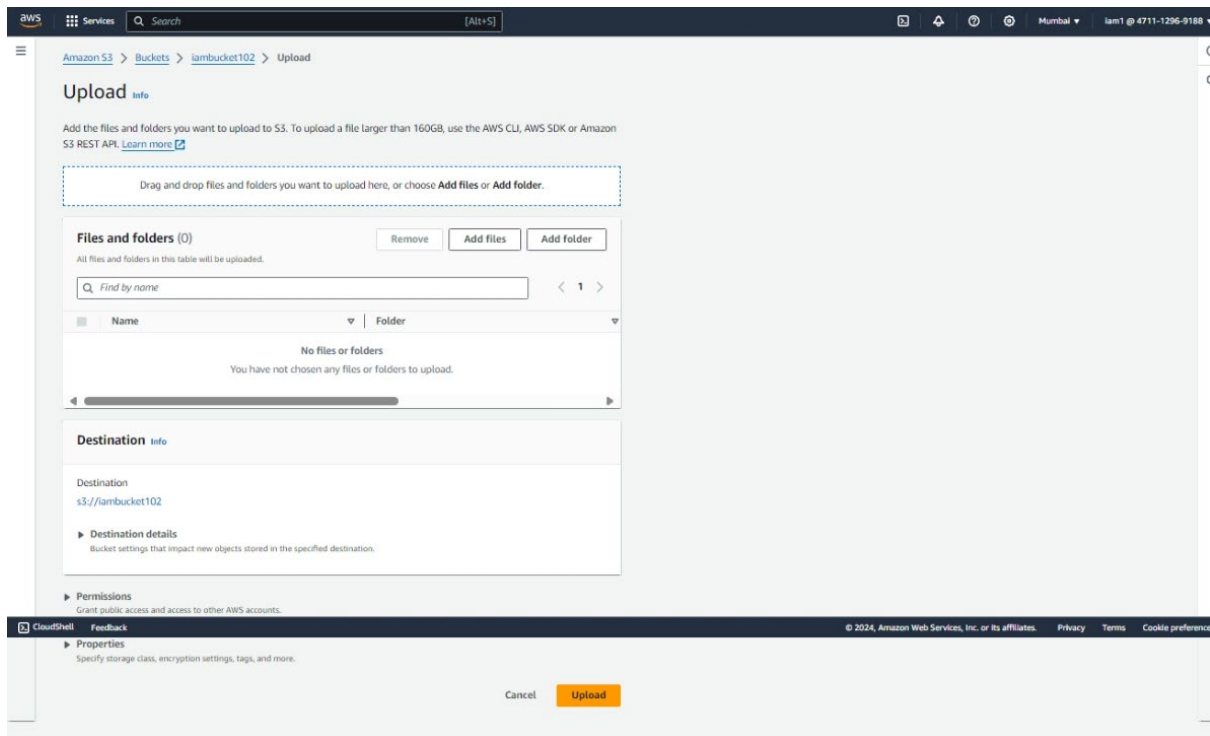
- Both buckets are visible from IAM user 1.



- Now let us upload a file into 1<sup>st</sup> bucket(iambucket102) from IAM user 1 and view it from IAM user 2.



- After clicking Upload button, it is redirected to the below page:



ServicesSearch[Alt+S]

Amazon S3> Buckets> iambucket102> 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 SDK 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 (1 Total, 6.0 KB)

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder
<input type="checkbox"/>	iamtask.csv	-

Destination

Info

Destination

s3://iambucket102

Destination details

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

Permissions

Grant public access and access to other AWS accounts.

Properties

aws

Services

Search

[Alt+S]

Amazon S3

>

Buckets

>

iambucket102

iambucket102

Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (1) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

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

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	iamtask.csv	csv	August 26, 2024, 10:41:46 (UTC+05:30)	6.0 KB	Standard

- Now let's upload another file to 2<sup>nd</sup> bucket using same procedure from 1<sup>st</sup> IAM user (iam1) only.

The screenshot displays the AWS Management Console interface for the 'iambucket202' bucket. The top navigation bar shows the AWS logo, 'Services', a search bar, and the user 'iam1 @ 4711-1296-9188'. The breadcrumb trail indicates the path: Amazon S3 > Buckets > iambucket202. The bucket's 'Objects' tab is selected, showing a list of objects. A message states 'No objects' and 'You don't have any objects in this bucket.' with an 'Upload' button. Below this, a green banner indicates 'Upload succeeded' with a 'View details below' link. The 'Upload: status' section shows a summary of the upload: Destination 's3://iambucket202', Succeeded '1 file, 5.0 B (100.00%)', and Failed '0 files, 0 B (0%)'. The 'Files and folders' tab is selected, showing a table with one file: 'amogh.csv' (text/csv, 5.0 B, Status: Succeeded).

Amazon S3 > Buckets > iambucket202

iambucket202 info

Objects (0) info

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

No objects  
You don't have any objects in this bucket.

Upload

Upload succeeded  
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination: s3://iambucket202

Succeeded: 1 file, 5.0 B (100.00%)

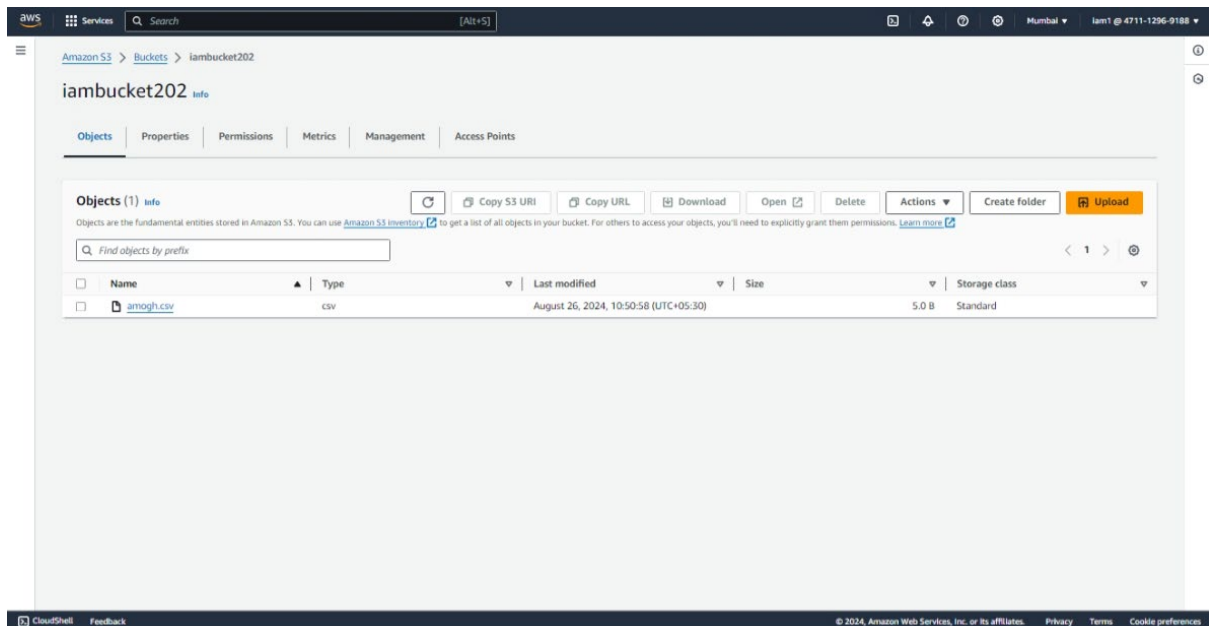
Failed: 0 files, 0 B (0%)

Files and folders (1 Total, 5.0 B)

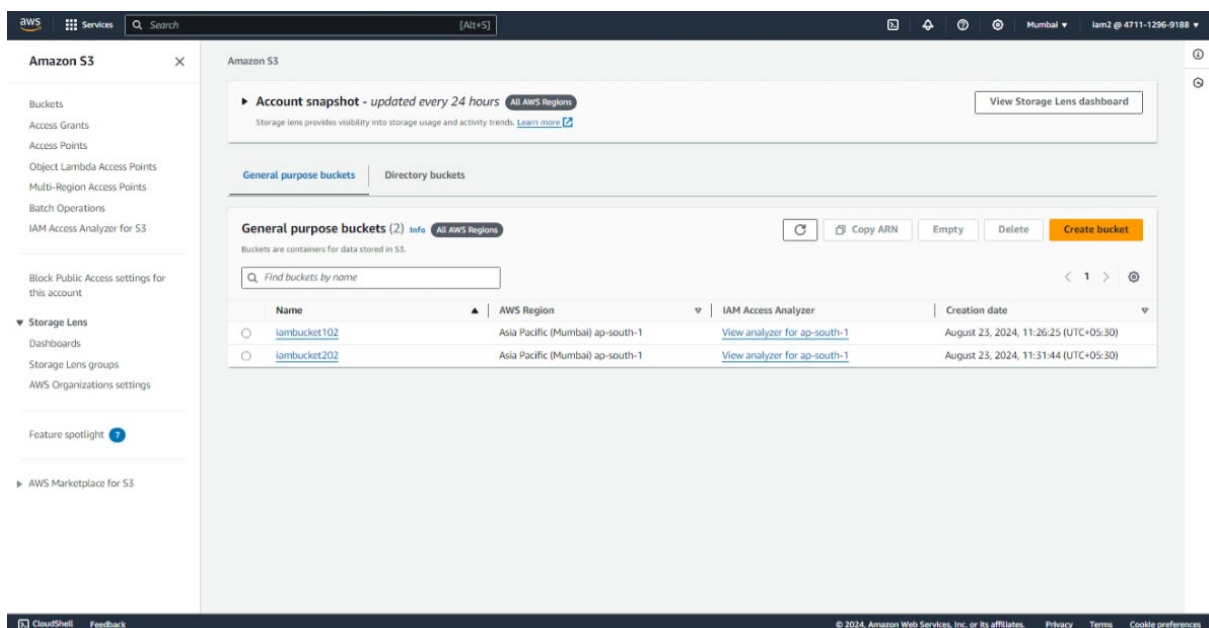
Find by name

Name	Folder	Type	Size	Status	Error
<a href="#">amogh.csv</a>	-	text/csv	5.0 B	Succeeded	-

- Now we uploaded a different file into 2<sup>nd</sup> bucket (iambucket202).

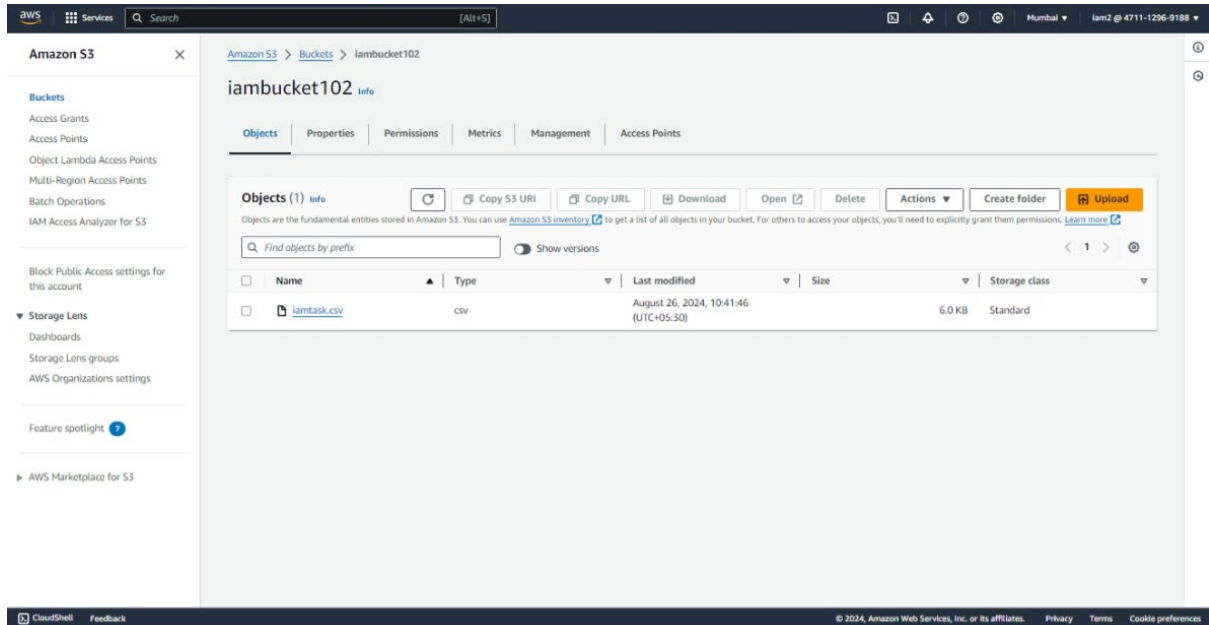


- We signed-in as iam2(IAM user 2) and we can see both buckets visible in S3 tab.

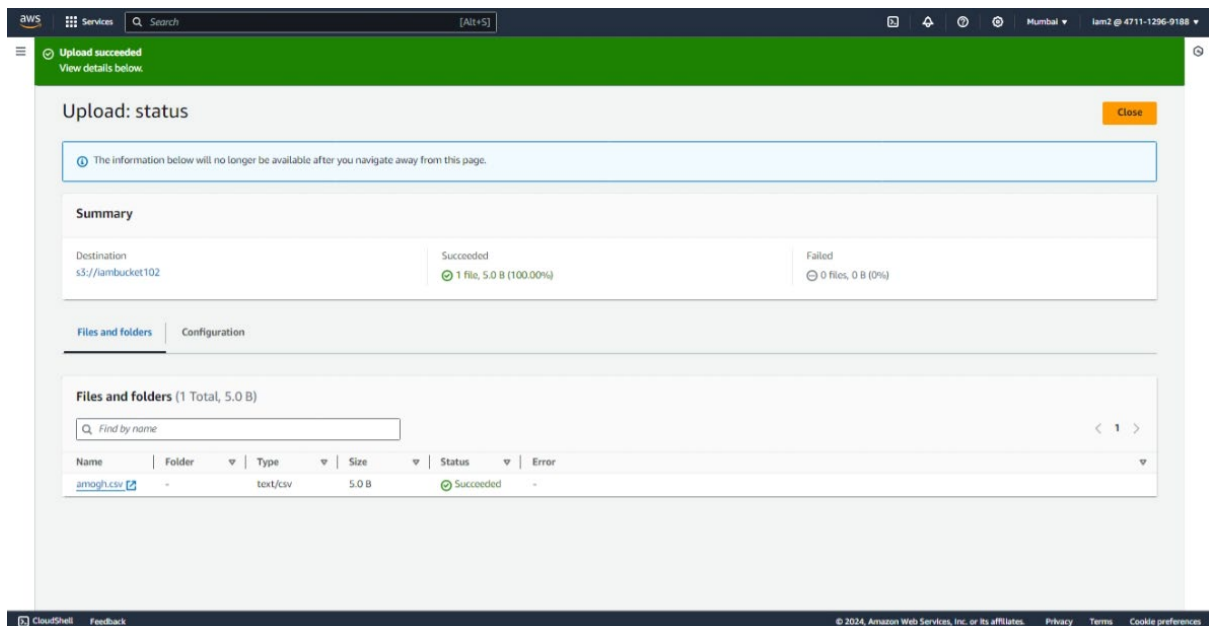




- We can view the file uploaded in 1<sup>st</sup> bucket too



- Now let's upload a file into 1<sup>st</sup> bucket



- File upload is successful.

The screenshot shows the Amazon S3 console interface for a bucket named 'iambucket102'. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > iambucket102. The bucket name 'iambucket102' is displayed with an 'info' link. Below this, there are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is active, showing a list of objects. Above the list, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar labeled 'Find objects by prefix' and a 'Show versions' toggle are also present. The object list table has columns for Name, Type, Last modified, Size, and Storage class. Two objects are listed: 'amogh.csv' (5.0 B, Standard) and 'iamtask.csv' (6.0 KB, Standard). The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. (© 2024).

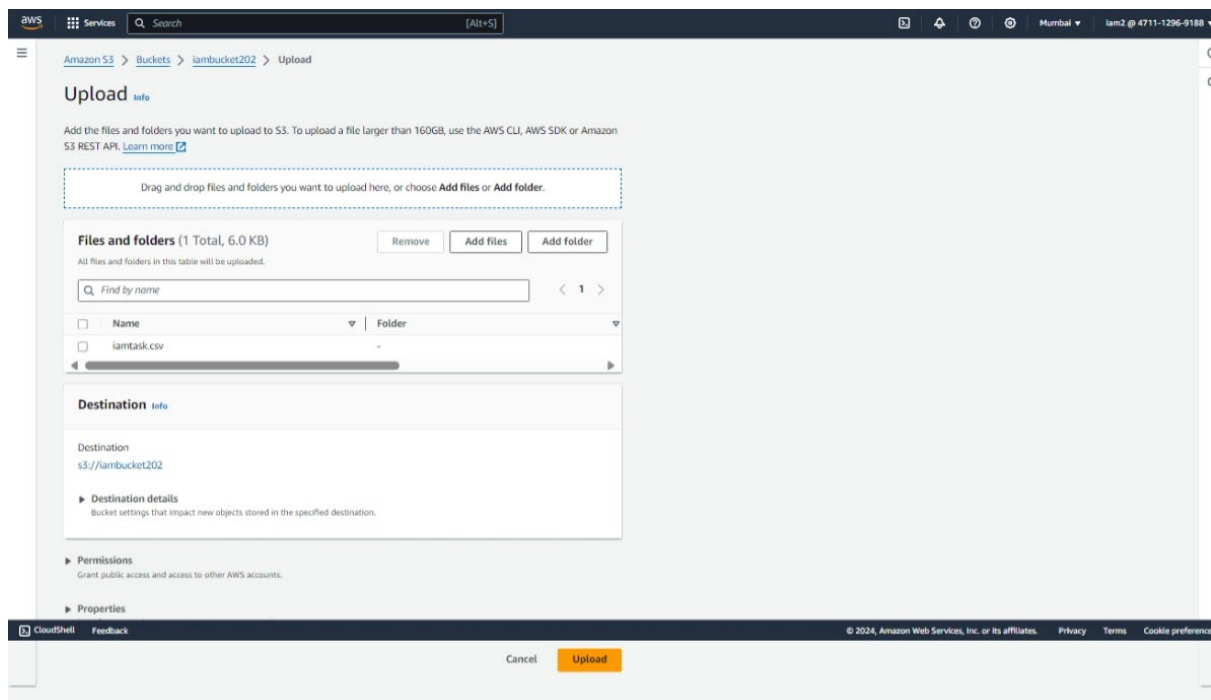
Name	Type	Last modified	Size	Storage class
<a href="#">amogh.csv</a>	CSV	August 26, 2024, 11:13:47 (UTC+05:30)	5.0 B	Standard
<a href="#">iamtask.csv</a>	CSV	August 26, 2024, 10:41:46 (UTC+05:30)	6.0 KB	Standard

- Now we'll go to 2<sup>nd</sup> bucket

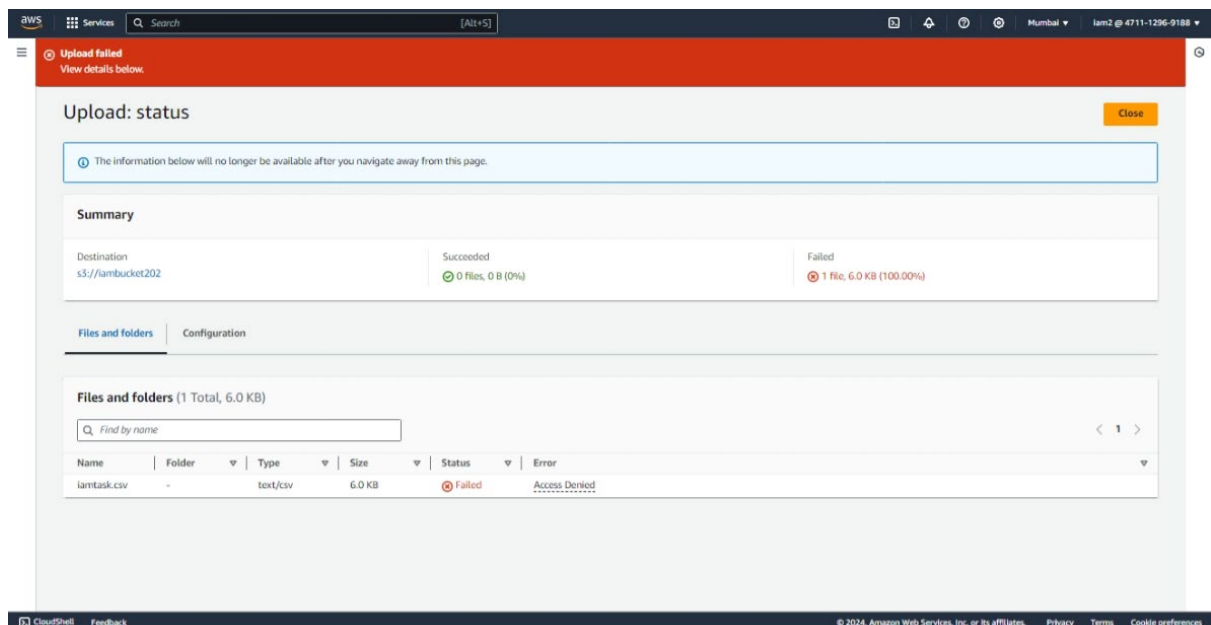
The screenshot shows the Amazon S3 console interface for a bucket named 'iambucket202'. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > iambucket202. The bucket name 'iambucket202' is displayed with an 'info' link. Below this, there are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is active, showing a list of objects. Above the list, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar labeled 'Find objects by prefix' and a 'Show versions' toggle are also present. The object list table has columns for Name, Type, Last modified, Size, and Storage class. One object is listed: 'amogh.csv' (5.0 B, Standard). The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. (© 2024).

Name	Type	Last modified	Size	Storage class
<a href="#">amogh.csv</a>	CSV	August 26, 2024, 10:50:58 (UTC+05:30)	5.0 B	Standard

- In 2<sup>nd</sup> bucket, all objects are visible. Now we'll try to upload an object



- Hence, when an object is uploaded into iambucket202(2<sup>nd</sup> bucket) from IAM user 2(iam2), it is giving error.



- Object is not uploaded in 2<sup>nd</sup> bucket from iam2(2<sup>nd</sup> IAM user)

