Codec Technologies : 1-Month Cloud Internship

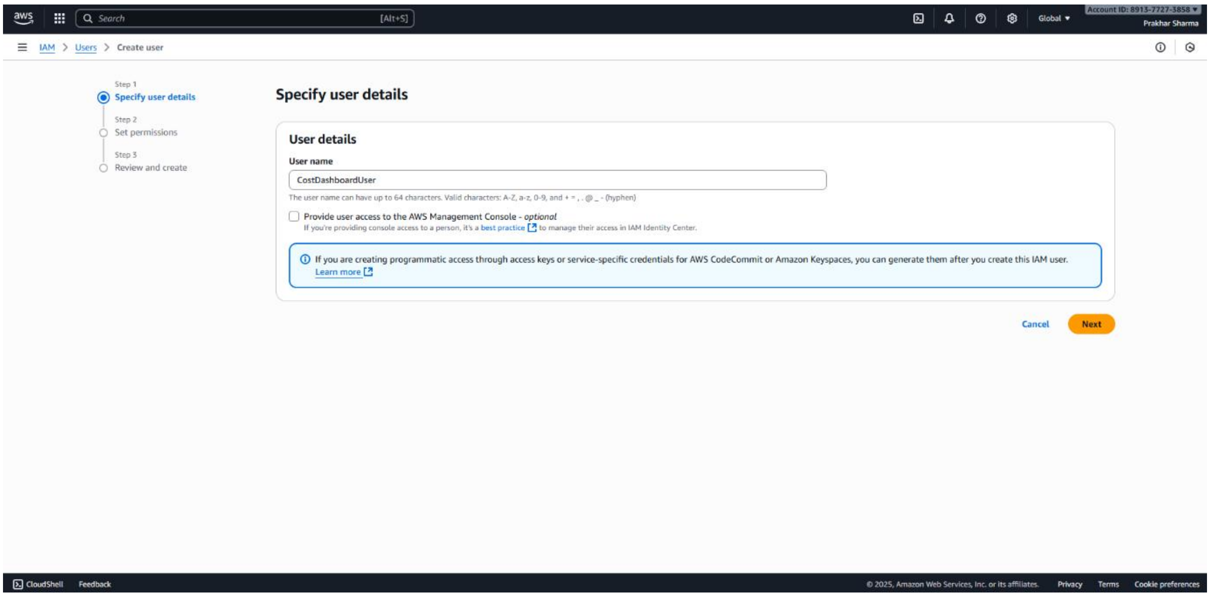
Name :Prakhar Anil Sharma

Major Project : Cloud Cost Optimization Dashboard

Objective: Build a dashboard to monitor and optimize cloud usage and costs.
Guidelines:

- Use AWS Cost Explorer API or Azure Cost Management.
- Visualize in Power BI or Grafana.
- Add budget alerts and optimization tips.

Step – 1] Go to AWS Management Console and log in.
Step – 2] Search IAM, At left navigation menu click on user.
Step – 3] Click create user.
Step – 4] Name the user.
Step – 5] Click next.



Step 1
● Specify user details
Step 2
○ Set permissions
Step 3
○ Review and create

Specify user details

User details

User name

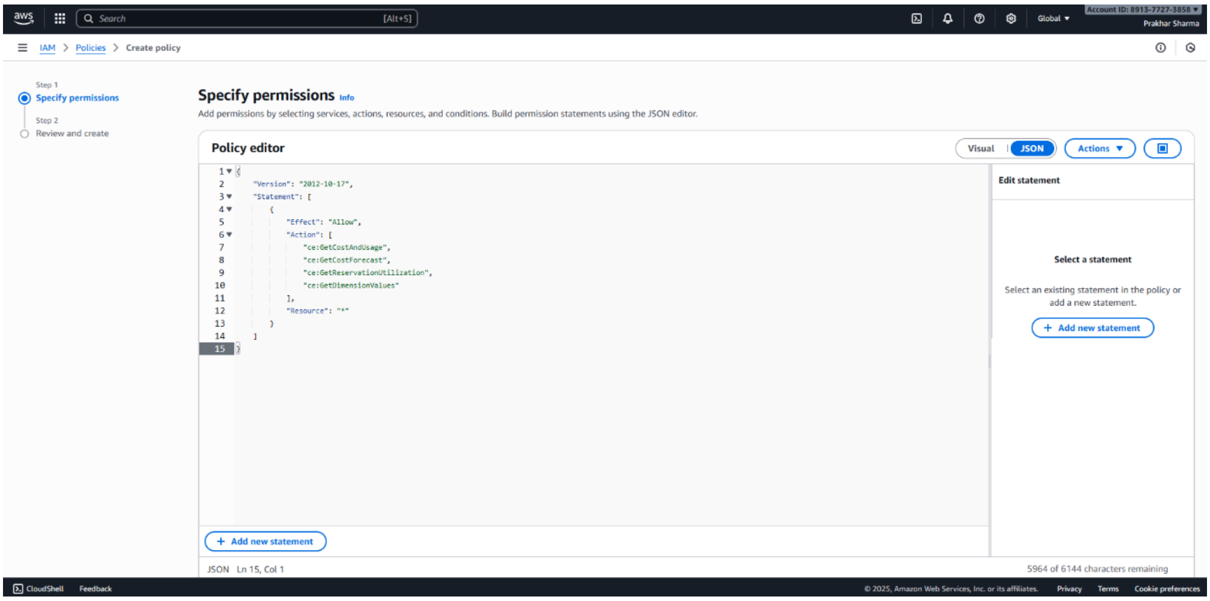
CostDashboardUser

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.

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

Step – 6] In the Navigation menu, Click on policy and open it in new tab.
Step – 7] Create policy.
Step – 8] Write the policy you want or as shown below .



Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ce:GetCostAndUsage",
                "ce:GetCostForecast",
                "ce:GetReservationUtilization",
                "ce:GetDimensionValues"
            ],
            "Resource": "*"
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

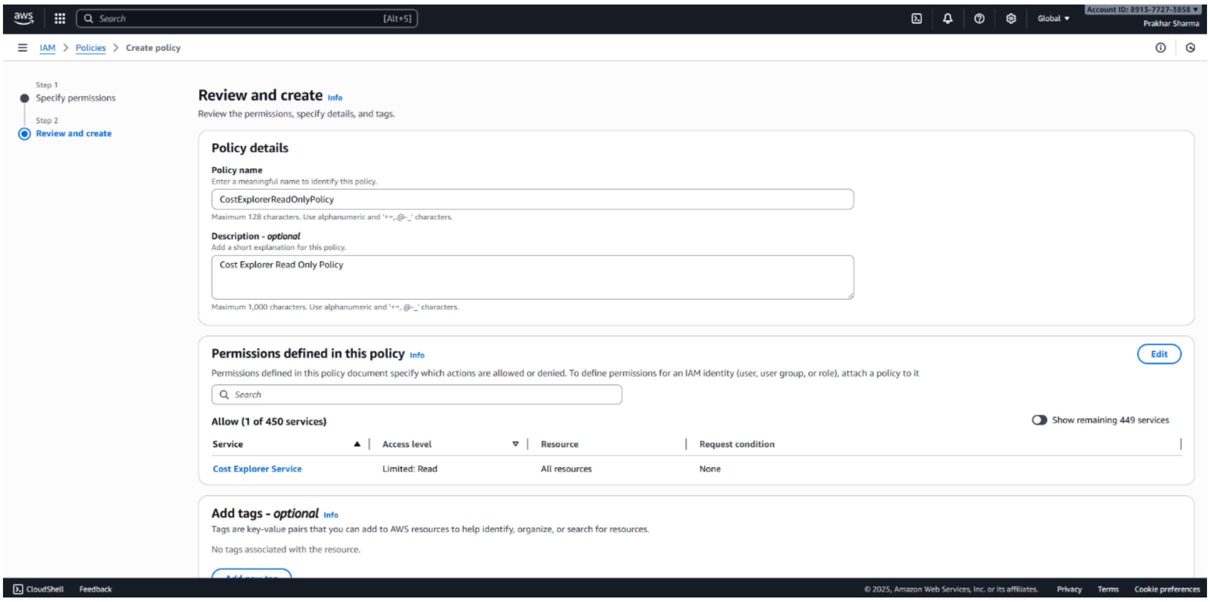
JSON Ln 15, Col 1

5964 of 6144 characters remaining

Step – 9] Click save and next.

Step -10] Name the policy and its description.

Step – 11] Click create policy.



Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

CostExplorerReadOnlyPolicy

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

Description - *optional*

Add a short explanation for this policy.

Cost Explorer Read Only Policy

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

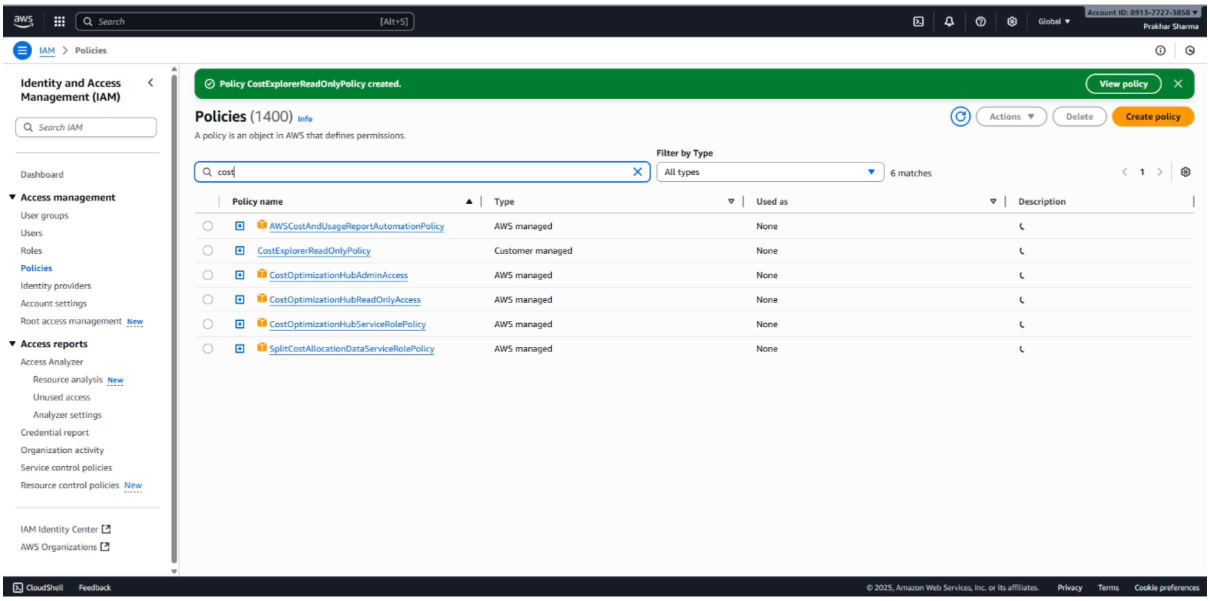
Allow (1 of 450 services)

Service	Access level	Resource	Request condition
Cost Explorer Service	Limited: Read	All resources	None

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.



Policy CostExplorerReadOnlyPolicy created.

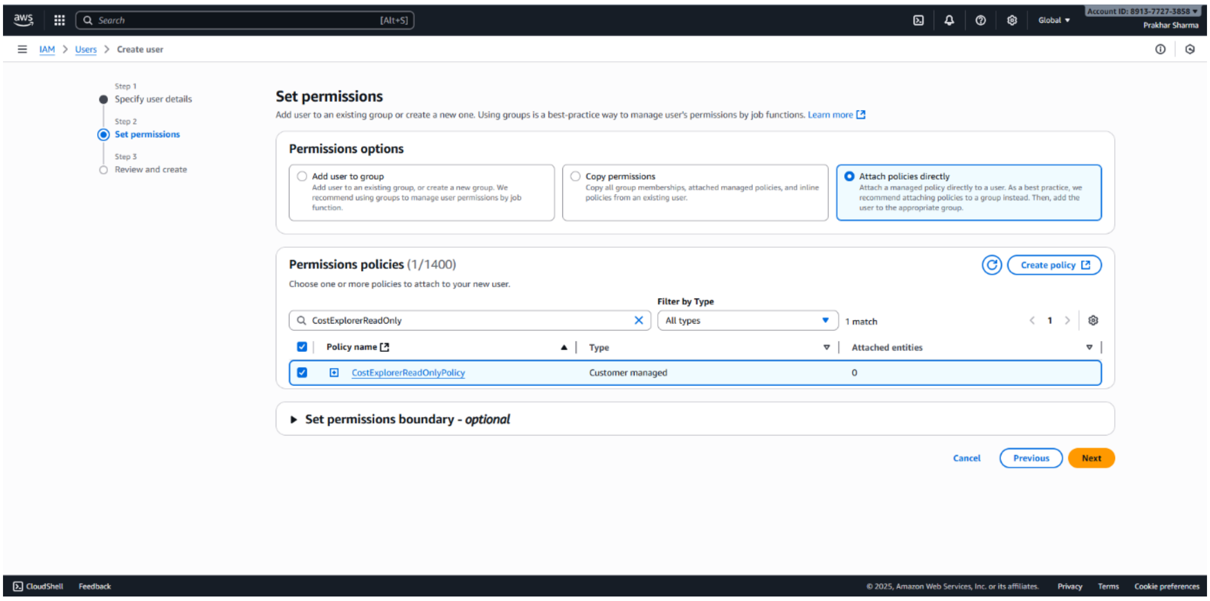
Policies (1400) Info

A policy is an object in AWS that defines permissions.

Search: cost — Filter by Type: All types — 6 matches

Policy name	Type	Used as	Description
AWSCostAndUsageReportAutomationPolicy	AWS managed	None	
CostExplorerReadOnlyPolicy	Customer managed	None	
CostOptimizationHubAdminAccess	AWS managed	None	
CostOptimizationHubReadOnlyAccess	AWS managed	None	
CostOptimizationHubServiceRolePolicy	AWS managed	None	
SplitCostAllocationDataServiceRolePolicy	AWS managed	None	

Step – 12] Now go back to IAM user you are creating, and attach the policy you created and click next .



Set permissions

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

Permissions options

○ Add user to group — Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions — Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly — Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1400)

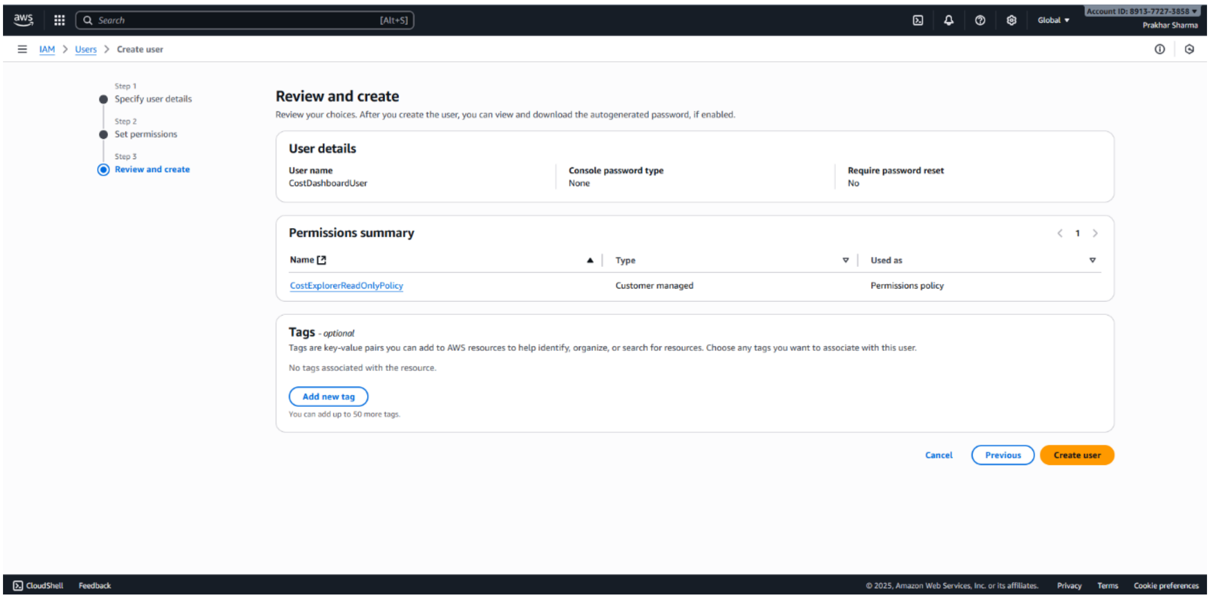
Choose one or more policies to attach to your new user.

Search: CostExplorerReadOnly — Filter by Type: All types — 1 match

☑	Policy name	Type	Attached entities
☑	CostExplorerReadOnlyPolicy	Customer managed	0

▸ Set permissions boundary - *optional*

Step -13] Click create user .



Review and create

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

User details

User name	Console password type	Require password reset
CostDashboardUser	None	No

Permissions summary

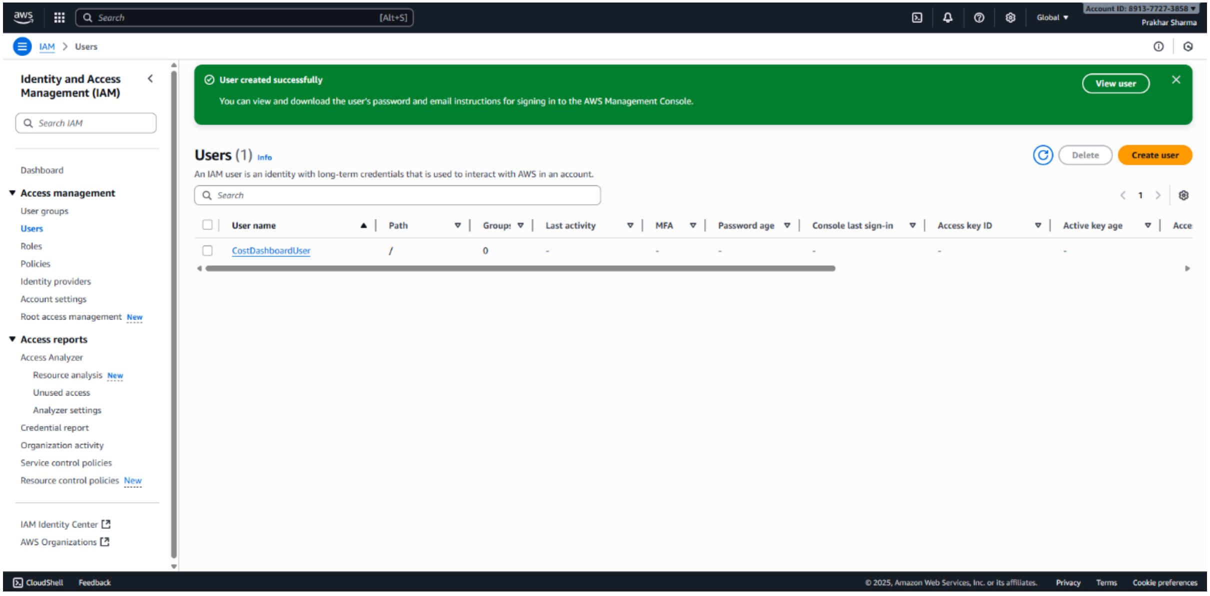
Name	Type	Used as
CostExplorerReadOnlyPolicy	Customer managed	Permissions policy

Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.



User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

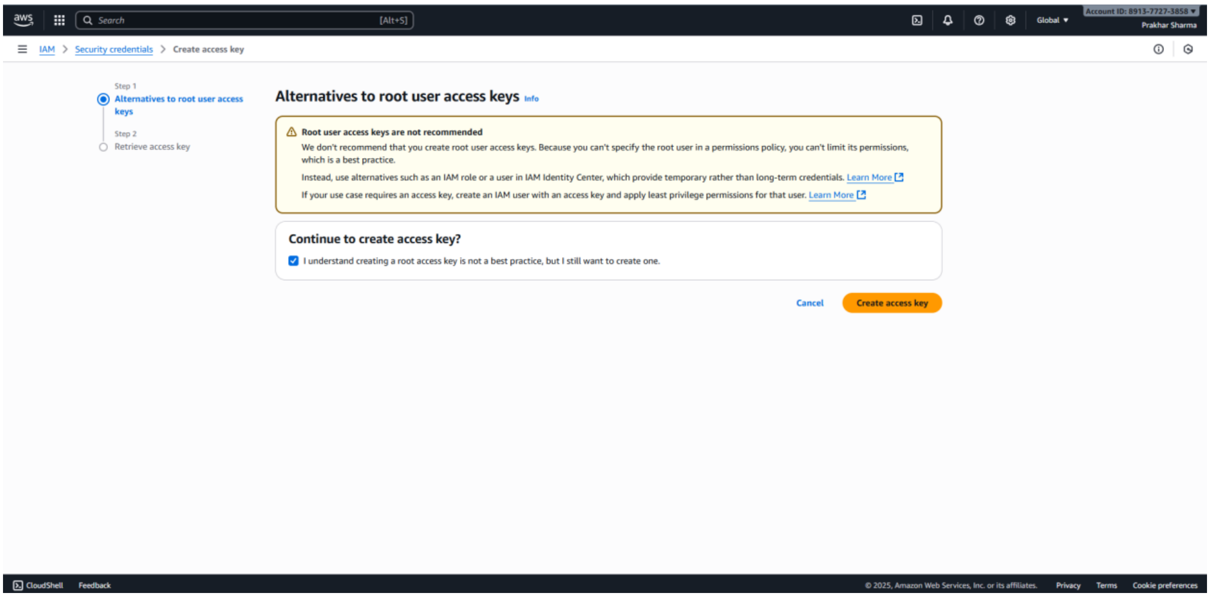
Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Acce
□	CostDashboardUser	/	0	-	-	-	-	-	-	

Step – 14] Now go to Security credential at top right corner.

Step – 15] Create access key and secret key , and save it in notepad .



Search [Alt+S]
Global
Account ID: 8913-7727-3858
Prakhar Sharma
IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Alternatives to root user access keys Info

Root user access keys are not recommended

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

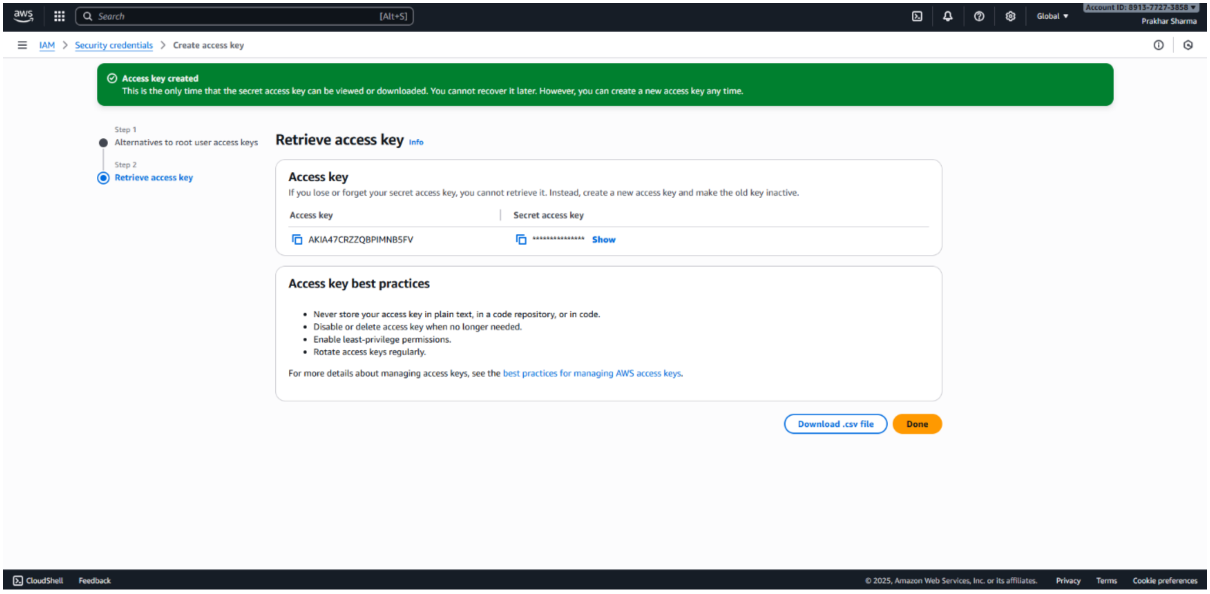
Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. Learn More

Continue to create access key?

☑ I understand creating a root access key is not a best practice, but I still want to create one.

Cancel | Create access key



Search [Alt+S]
Global
Account ID: 8913-7727-3858
Prakhar Sharma
IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access key Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA47CRZZQBPIMNB5FV	*************** Show

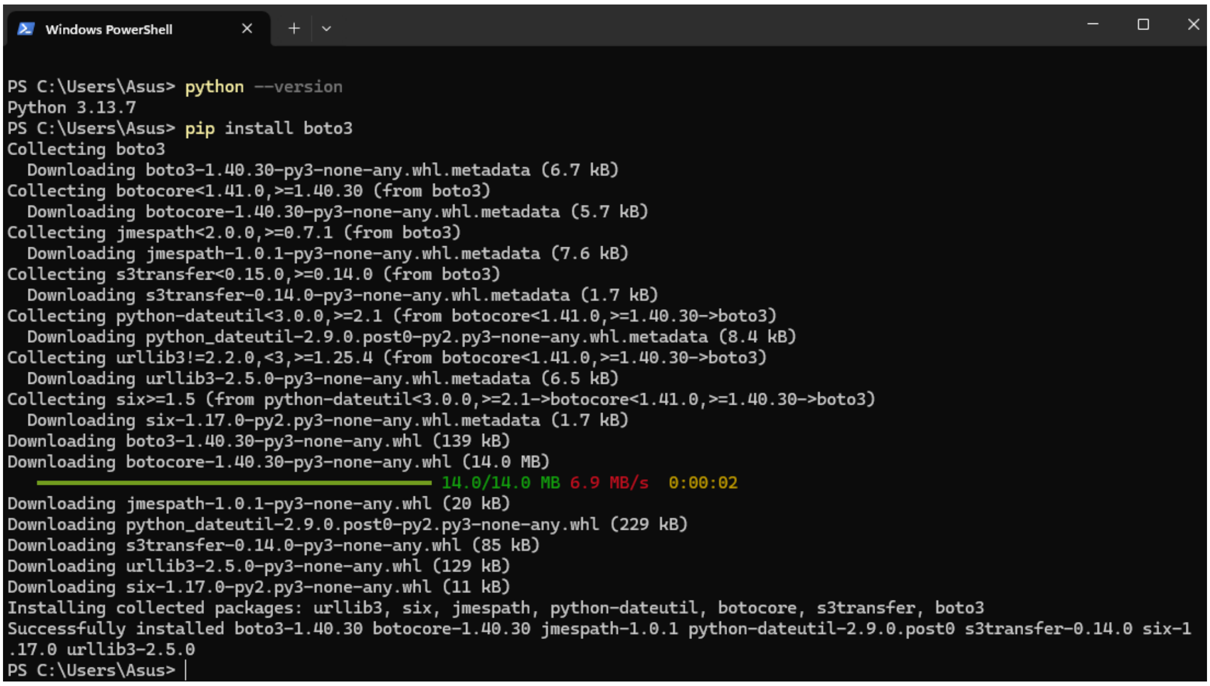
Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

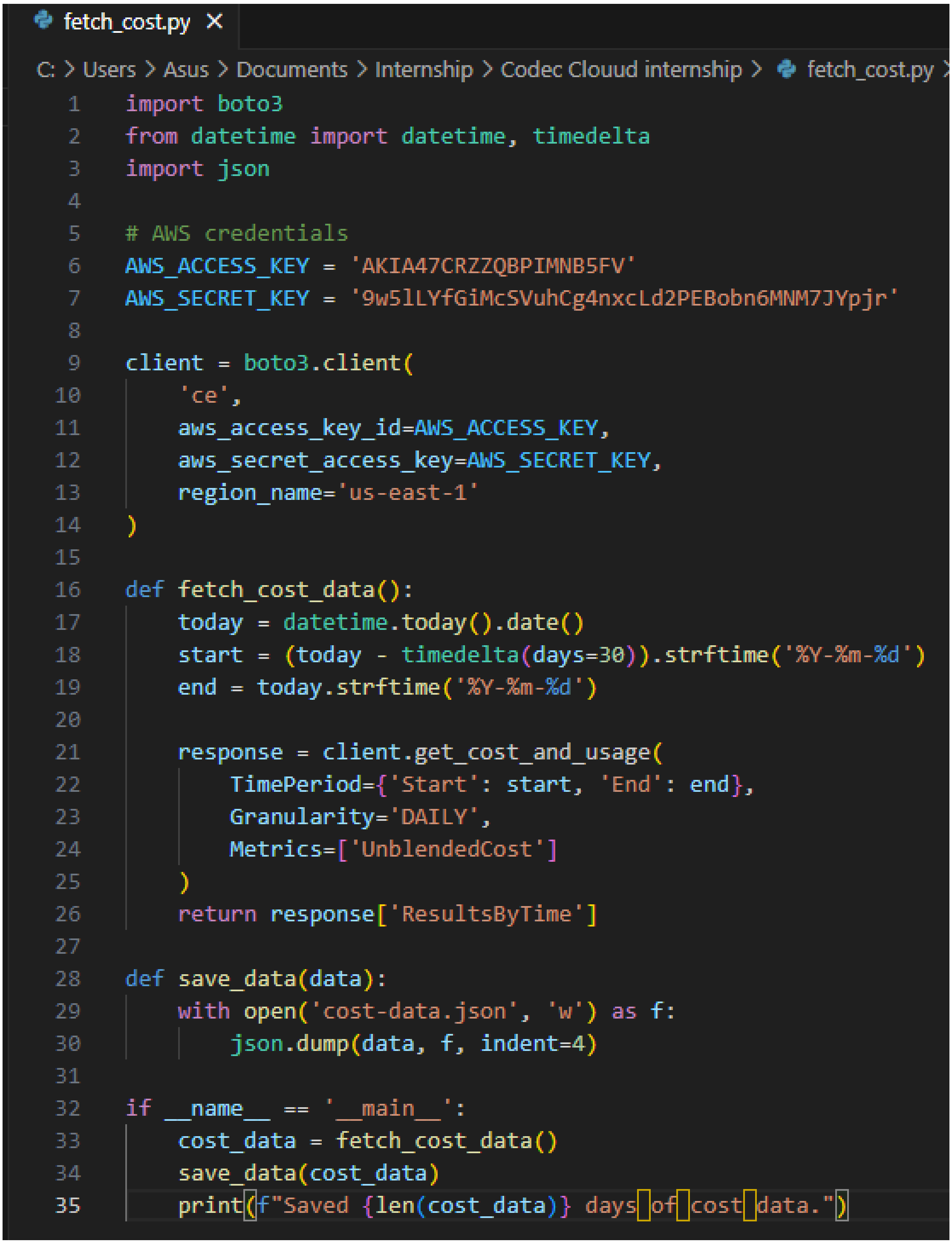
Download .csv file | Done

Step – 16] Now Install Python and boto3 on your machine .



```
PS C:\Users\Asus> python --version
Python 3.13.7
PS C:\Users\Asus> pip install boto3
Collecting boto3
  Downloading boto3-1.40.30-py3-none-any.whl.metadata (6.7 kB)
Collecting botocore<1.41.0,>=1.40.30 (from boto3)
  Downloading botocore-1.40.30-py3-none-any.whl.metadata (5.7 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.15.0,>=0.14.0 (from boto3)
  Downloading s3transfer-0.14.0-py3-none-any.whl.metadata (1.7 kB)
Collecting python-dateutil<3.0.0,>=2.1 (from botocore<1.41.0,>=1.40.30->boto3)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting urllib3!=2.2.0,<3,>=1.25.4 (from botocore<1.41.0,>=1.40.30->boto3)
  Downloading urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.41.0,>=1.40.30->boto3)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading boto3-1.40.30-py3-none-any.whl (139 kB)
Downloading botocore-1.40.30-py3-none-any.whl (14.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.0/14.0 MB 6.9 MB/s  0:00:02
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading s3transfer-0.14.0-py3-none-any.whl (85 kB)
Downloading urllib3-2.5.0-py3-none-any.whl (129 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, jmespath, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.40.30 botocore-1.40.30 jmespath-1.0.1 python-dateutil-2.9.0.post0 s3transfer-0.14.0 six-1
.17.0 urllib3-2.5.0
PS C:\Users\Asus> |
```

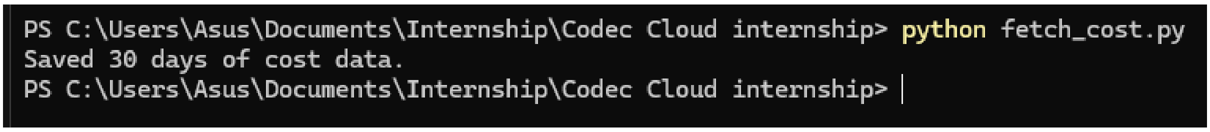
Step – 17] Now write a python script for fetching the cost for AWS cost explorer ,like shown below .



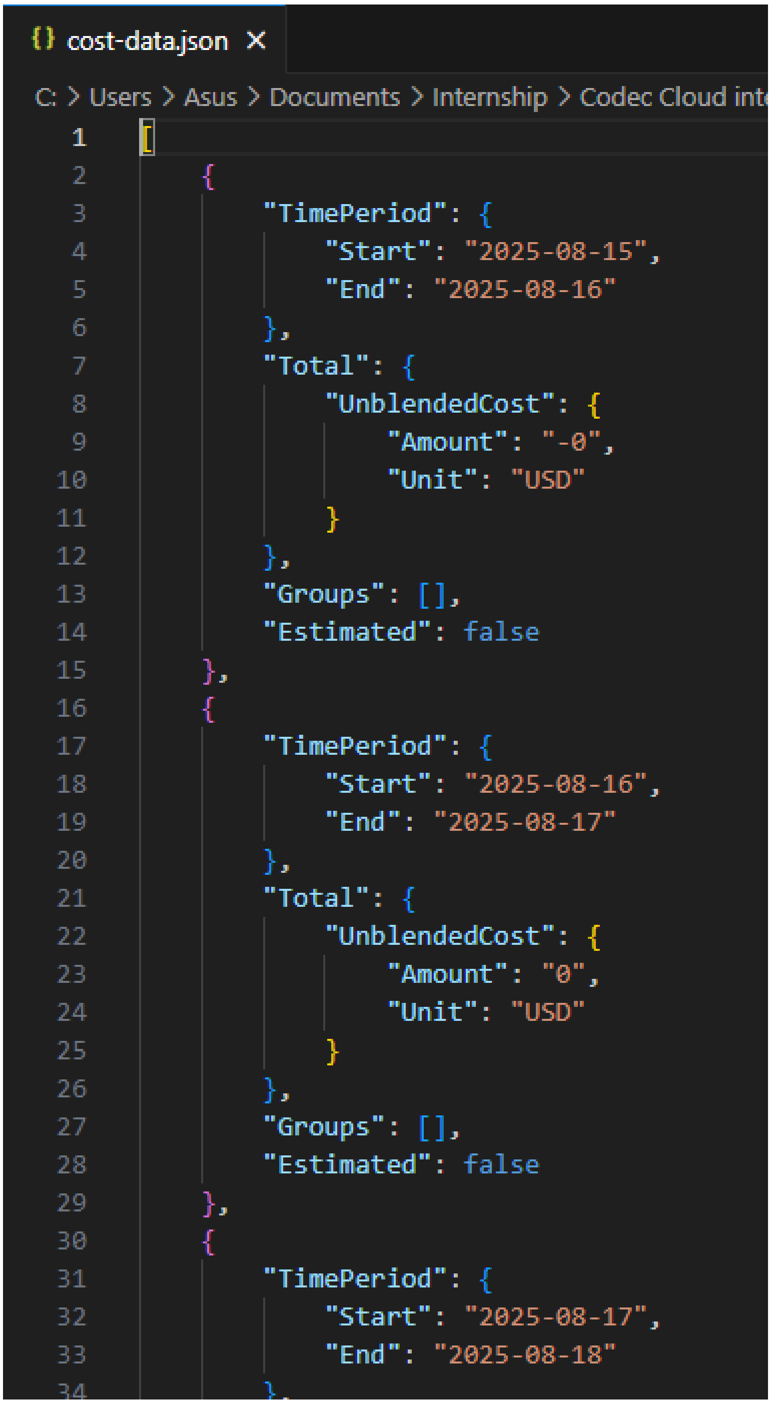
fetch_cost.py ✕
C: > Users > Asus > Documents > Internship > Codec Clouud internship > fetch_cost.py >

```python
import boto3
from datetime import datetime, timedelta
import json

# AWS credentials
AWS_ACCESS_KEY = 'AKIA47CRZZQBPIMNB5FV'
AWS_SECRET_KEY = '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'

client = boto3.client(
    'ce',
    aws_access_key_id=AWS_ACCESS_KEY,
    aws_secret_access_key=AWS_SECRET_KEY,
    region_name='us-east-1'
)

def fetch_cost_data():
    today = datetime.today().date()
    start = (today - timedelta(days=30)).strftime('%Y-%m-%d')
    end = today.strftime('%Y-%m-%d')

    response = client.get_cost_and_usage(
        TimePeriod={'Start': start, 'End': end},
        Granularity='DAILY',
        Metrics=['UnblendedCost']
    )
    return response['ResultsByTime']

def save_data(data):
    with open('cost-data.json', 'w') as f:
        json.dump(data, f, indent=4)

if __name__ == '__main__':
    cost_data = fetch_cost_data()
    save_data(cost_data)
    print(f"Saved {len(cost_data)} days of cost data.")
```

Step -18] Now run the script and you will get the output like this which will generate a json file named cost-data.json .



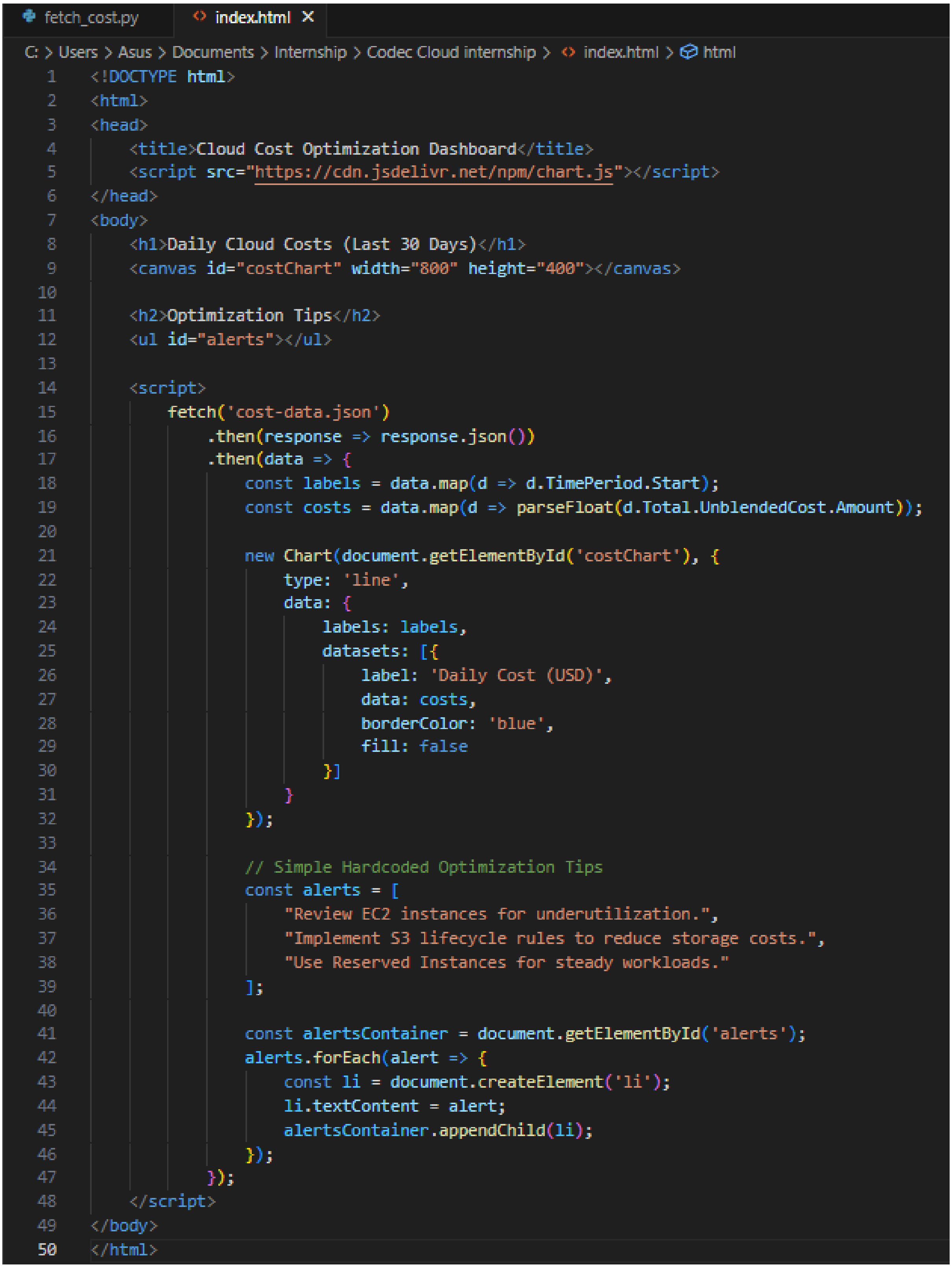
```
PS C:\Users\Asus\Documents\Internship\Codec Cloud internship> python fetch_cost.py
Saved 30 days of cost data.
PS C:\Users\Asus\Documents\Internship\Codec Cloud internship> 
```



cost-data.json
C: > Users > Asus > Documents > Internship > Codec Cloud int

```
[
    {
        "TimePeriod": {
            "Start": "2025-08-15",
            "End": "2025-08-16"
        },
        "Total": {
            "UnblendedCost": {
                "Amount": "-0",
                "Unit": "USD"
            }
        },
        "Groups": [],
        "Estimated": false
    },
    {
        "TimePeriod": {
            "Start": "2025-08-16",
            "End": "2025-08-17"
        },
        "Total": {
            "UnblendedCost": {
                "Amount": "0",
                "Unit": "USD"
            }
        },
        "Groups": [],
        "Estimated": false
    },
    {
        "TimePeriod": {
            "Start": "2025-08-17",
            "End": "2025-08-18"
        },
```

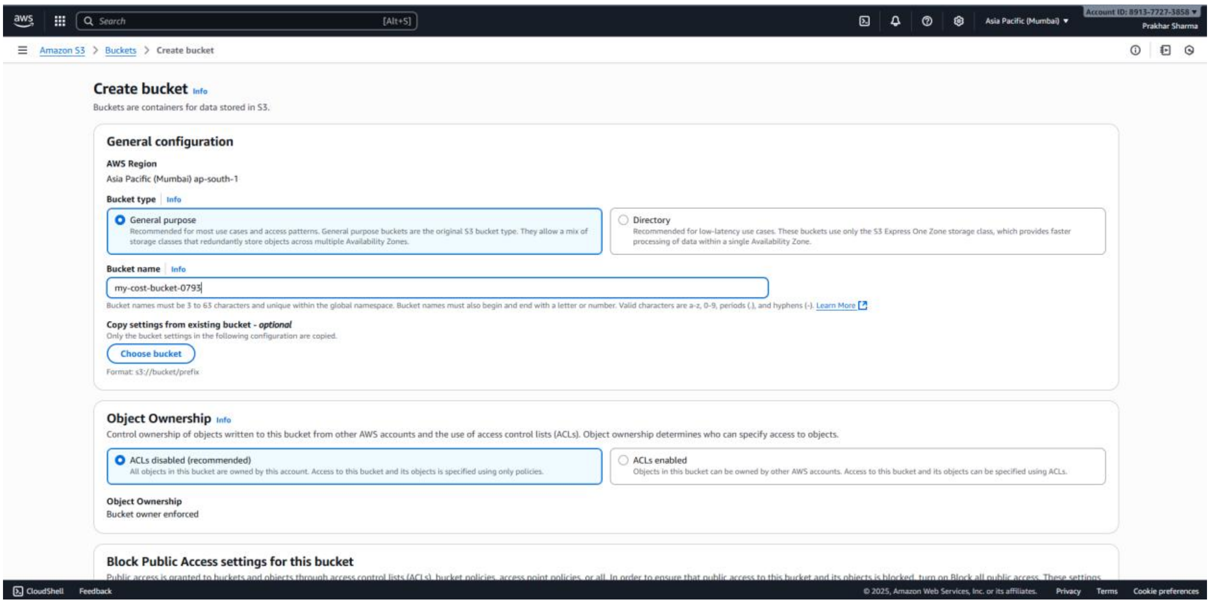
Step – 19] Now write a html page code for the dashboard of the project I want to make and the code will be like below .


fetch_cost.py
index.html ×
C: > Users > Asus > Documents > Internship > Codec Cloud internship > index.html > html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Cloud Cost Optimization Dashboard</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
    <h1>Daily Cloud Costs (Last 30 Days)</h1>
    <canvas id="costChart" width="800" height="400"></canvas>

    <h2>Optimization Tips</h2>
    <ul id="alerts"></ul>

    <script>
        fetch('cost-data.json')
            .then(response => response.json())
            .then(data => {
                const labels = data.map(d => d.TimePeriod.Start);
                const costs = data.map(d => parseFloat(d.Total.UnblendedCost.Amount));

                new Chart(document.getElementById('costChart'), {
                    type: 'line',
                    data: {
                        labels: labels,
                        datasets: [{
                            label: 'Daily Cost (USD)',
                            data: costs,
                            borderColor: 'blue',
                            fill: false
                        }]
                    }
                });

                // Simple Hardcoded Optimization Tips
                const alerts = [
                    "Review EC2 instances for underutilization.",
                    "Implement S3 lifecycle rules to reduce storage costs.",
                    "Use Reserved Instances for steady workloads."
                ];

                const alertsContainer = document.getElementById('alerts');
                alerts.forEach(alert => {
                    const li = document.createElement('li');
                    li.textContent = alert;
                    alertsContainer.appendChild(li);
                });
            });
    </script>
</body>
</html>
```

Step – 20] Now go to AWS and search S3 bucket and click create bucket .

Step – 21] Name the bucket .


aws
Search [Alt+S]
Asia Pacific (Mumbai) ▾
Account ID: 8913-7727-3858 ▾
Prakhar Sharma
Amazon S3 > Buckets > Create bucket
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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

my-cost-bucket-0793

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.

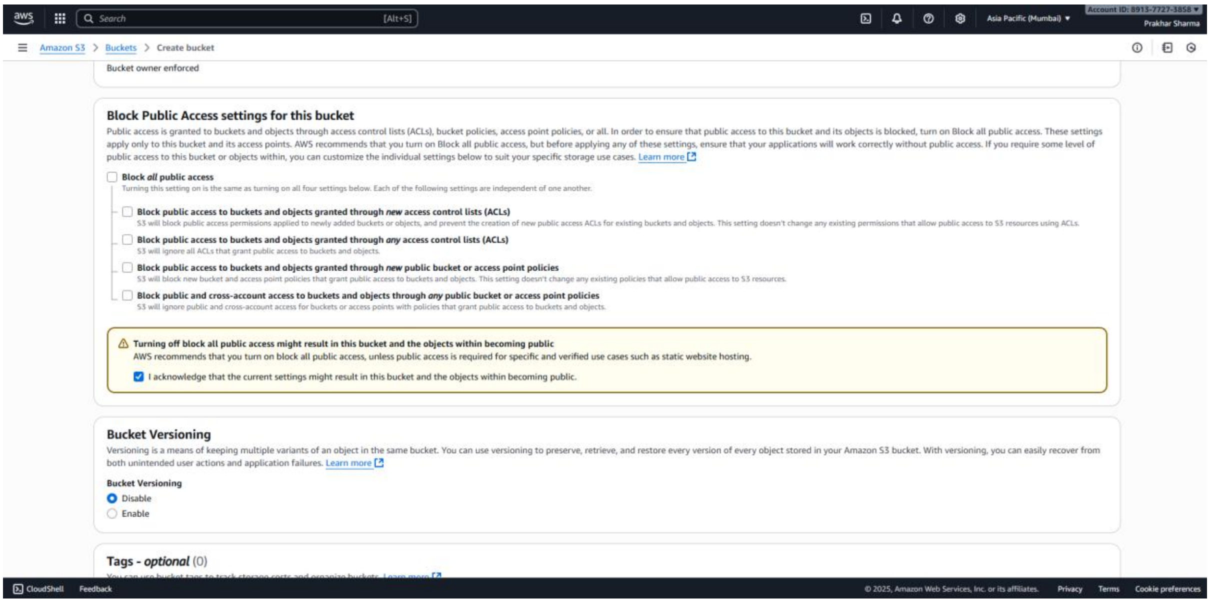
Object Ownership

Bucket owner enforced

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

Step – 22] Now uncheck the box of block public access and tick the acknowledgement box .


aws
Search [Alt+S]
Asia Pacific (Mumbai) ▾
Account ID: 8913-7727-3858 ▾
Prakhar Sharma
Amazon S3 > Buckets > Create bucket
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Bucket owner enforced

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

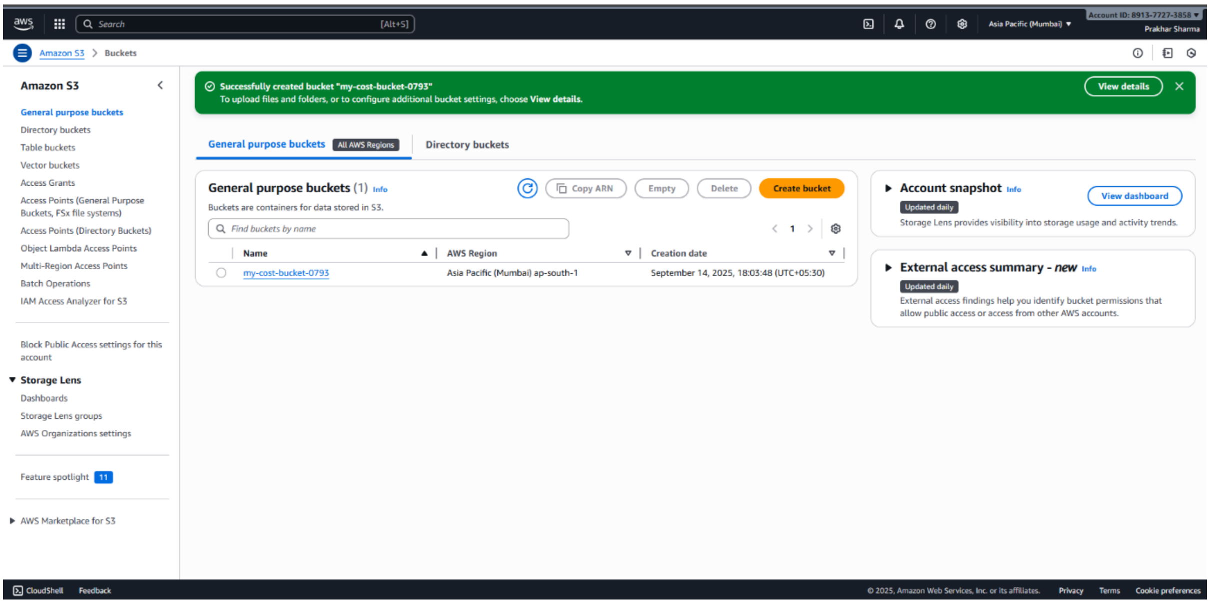
● Disable

○ Enable

Tags - *optional* (0)

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

Step – 23] Now click create bucket .



Successfully created bucket "my-cost-bucket-0793"
To upload files and folders, or to configure additional bucket settings, choose **View details.**

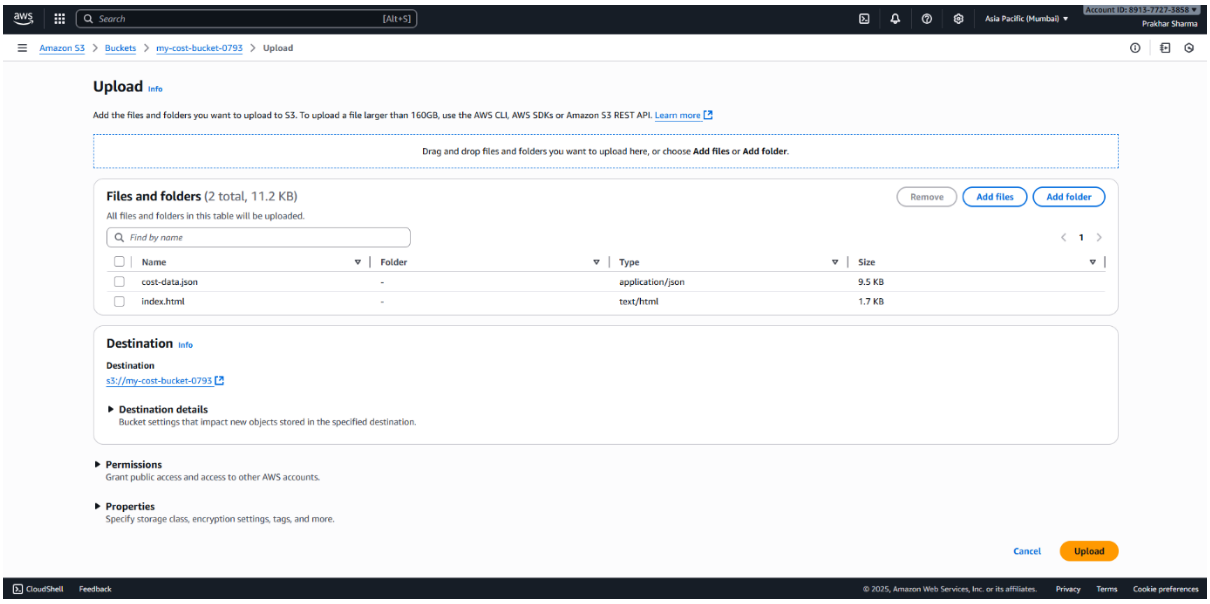
General purpose buckets (1)
Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
my-cost-bucket-0793	Asia Pacific (Mumbai) ap-south-1	September 14, 2025, 18:03:48 (UTC+05:30)

Account snapshot
Updated daily
Storage Lens provides visibility into storage usage and activity trends.

External access summary - *new*
Updated daily
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

Step – 24] Now click on the bucket, and upload the index.hrml and cost-data.json file .



Upload

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 (2 total, 11.2 KB)
All files and folders in this table will be uploaded.

Name	Folder	Type	Size
cost-data.json	-	application/json	9.5 KB
index.html	-	text/html	1.7 KB

Destination

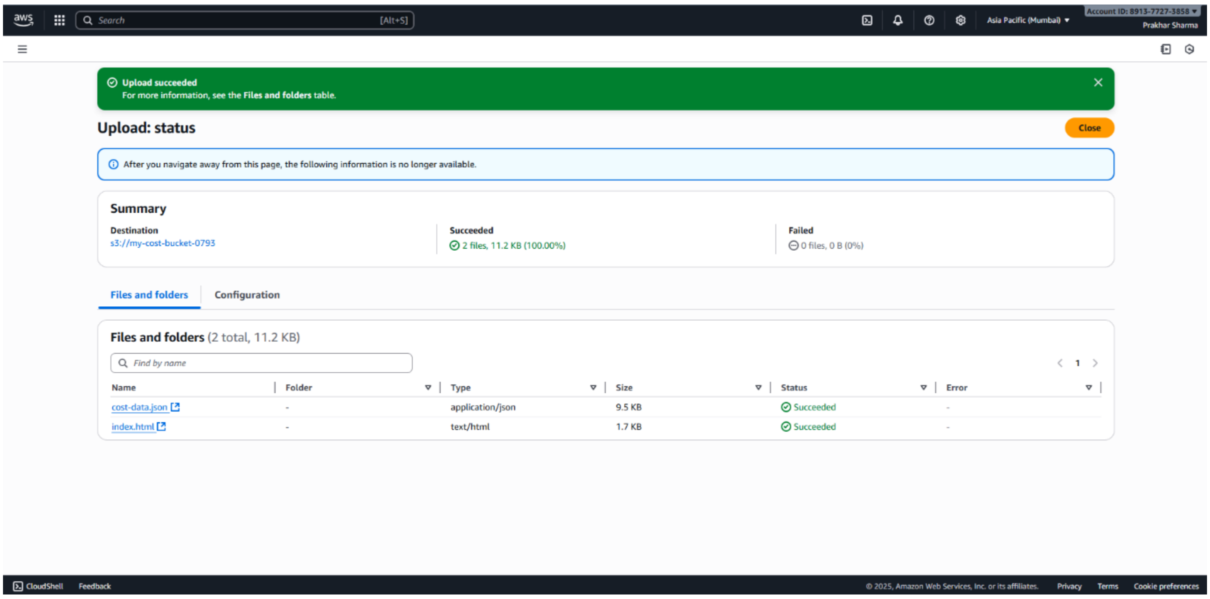
Destination
s3://my-cost-bucket-0793

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.

Amazon S3 > Buckets > my-cost-bucket-0793 > Upload
Remove
Add files
Add folder
Cancel
Upload
Account ID: 8913-7727-3858
Prakhar Sharma
Asia Pacific (Mumbai)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Upload succeeded
For more information, see the **Files and folders** table.

Upload: status

After you navigate away from this page, the following information is no longer available.

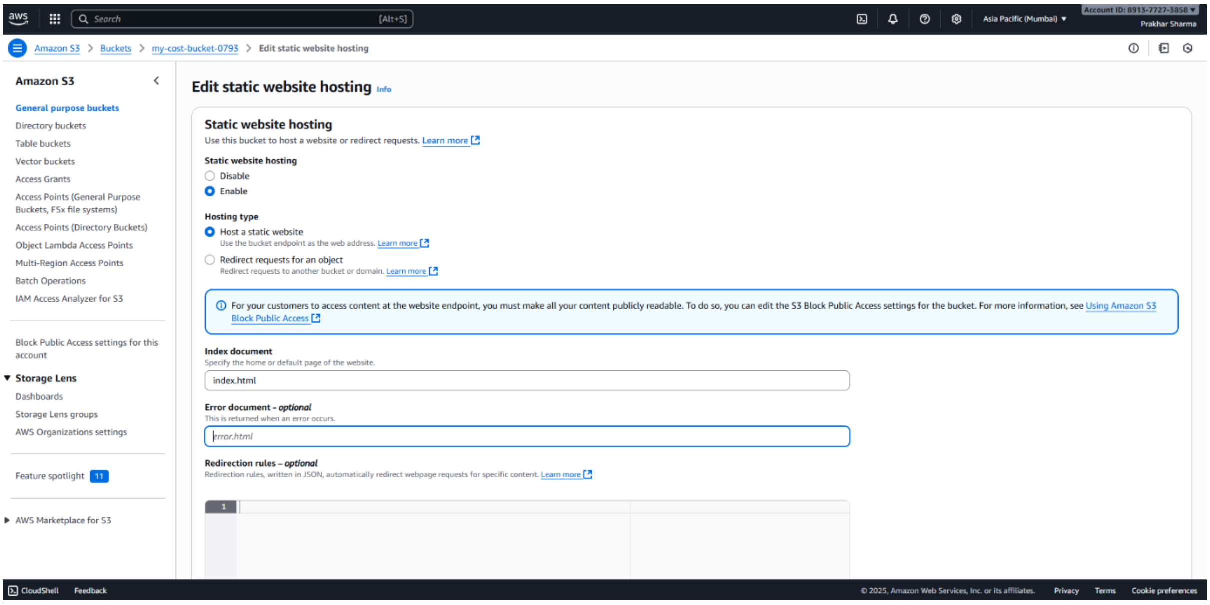
Summary

Destination	Succeeded	Failed
s3://my-cost-bucket-0793	✓ 2 files, 11.2 KB (100.00%)	0 files, 0 B (0%)

Files and folders (2 total, 11.2 KB)

Name	Folder	Type	Size	Status	Error
cost-data.json	-	application/json	9.5 KB	✓ Succeeded	-
index.html	-	text/html	1.7 KB	✓ Succeeded	-

Step – 25] Now go to permission tab and enable the static website hosting .
Step -26] Now in Index document write index.html as main page .
Step – 27[Click save.



aws
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8913-7727-3858
Prakhar Sharma
Amazon S3 > Buckets > my-cost-bucket-0793 > Edit static website hosting
Amazon S3
General purpose buckets
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
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Info
Learn more
1
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Edit static website hosting

Static website hosting
Use this bucket to host a website or redirect requests.

Static website hosting
○ Disable
● Enable

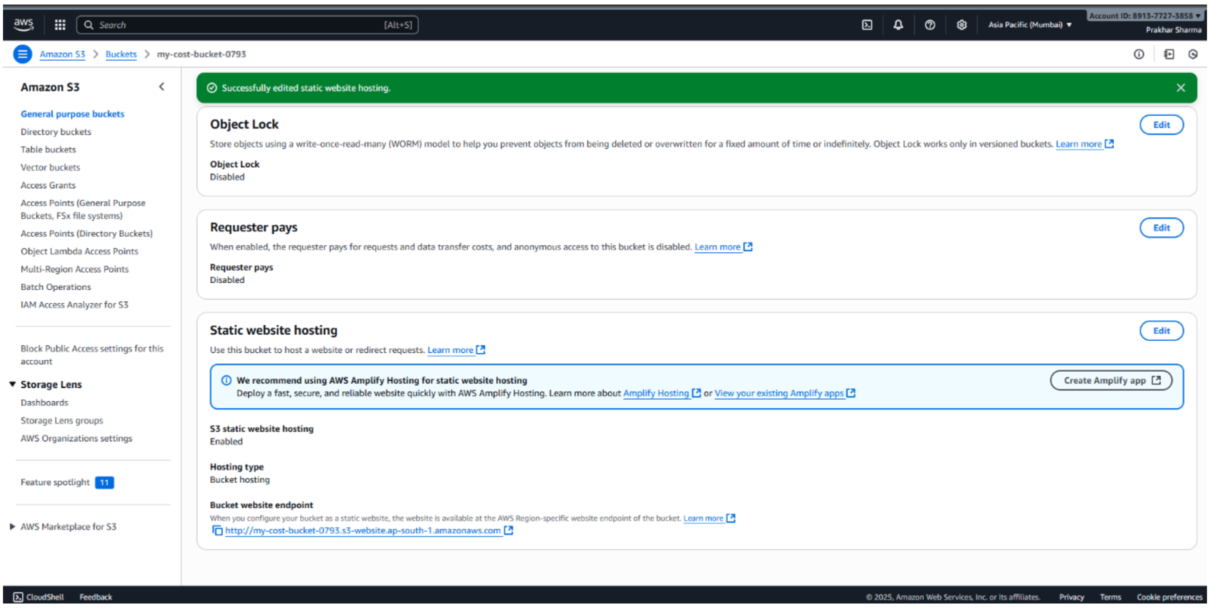
Hosting type
● Host a static website
Use the bucket endpoint as the web address.
○ Redirect requests for an object
Redirect requests to another bucket or domain.

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.


aws
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8913-7727-3858
Prakhar Sharma
Amazon S3 > Buckets > my-cost-bucket-0793
Amazon S3
General purpose buckets
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
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Edit
Create Amplify app
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Successfully edited static website hosting.

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

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

Requester pays: Disabled

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

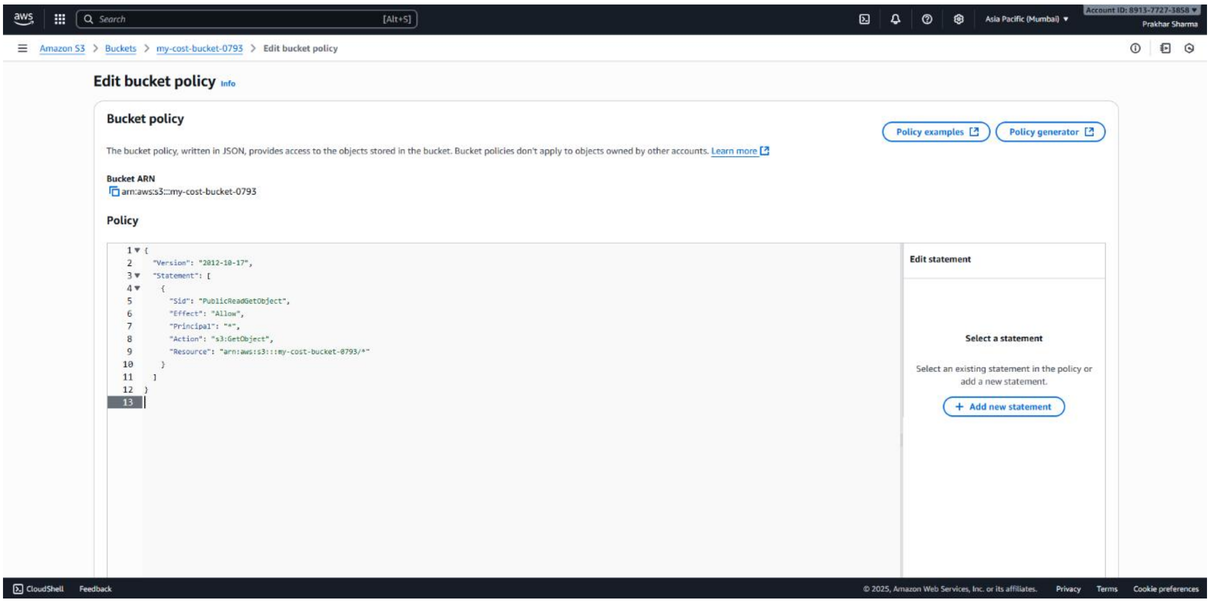
We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps

S3 static website hosting: Enabled

Hosting type: Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://my-cost-bucket-0793.s3-website.ap-south-1.amazonaws.com

Step – 28] Now go to properties tab, and write the policy as shown below by clicking on edit and save it after writing policy .



Edit bucket policy

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

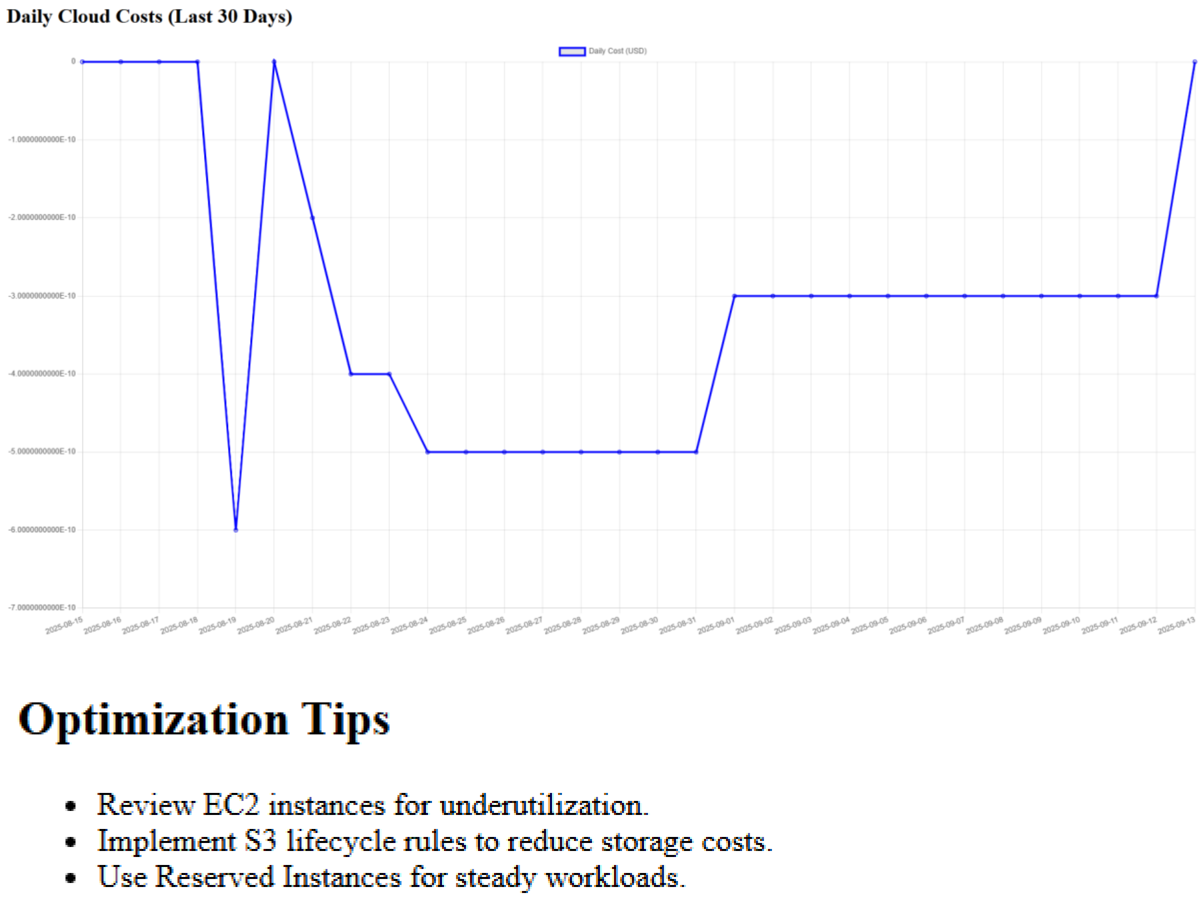
Bucket ARN
arn:aws:s3:::my-cost-bucket-0793

Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-cost-bucket-0793/*"
    }
  ]
}
```

Select an existing statement in the policy or add a new statement.

Step – 29] Now copy the static website link and paste it in new tab , you can see the cost dashboard.



Daily Cloud Costs (Last 30 Days)

Optimization Tips

- Review EC2 instances for underutilization.
- Implement S3 lifecycle rules to reduce storage costs.
- Use Reserved Instances for steady workloads.