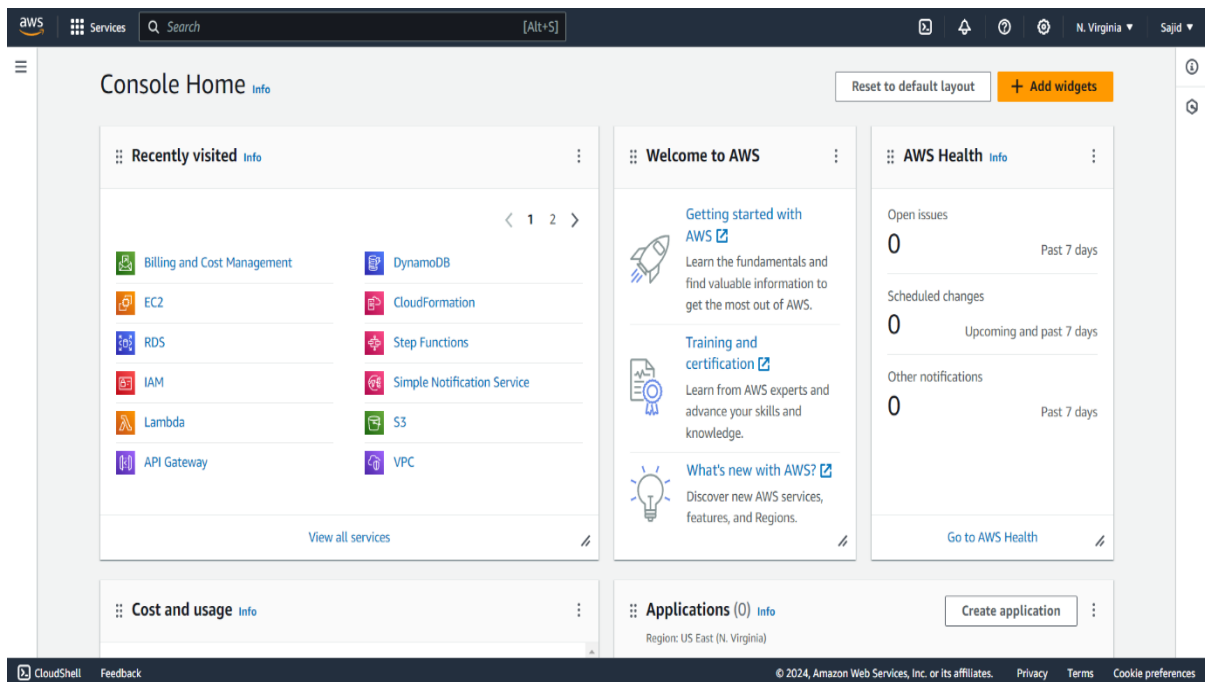
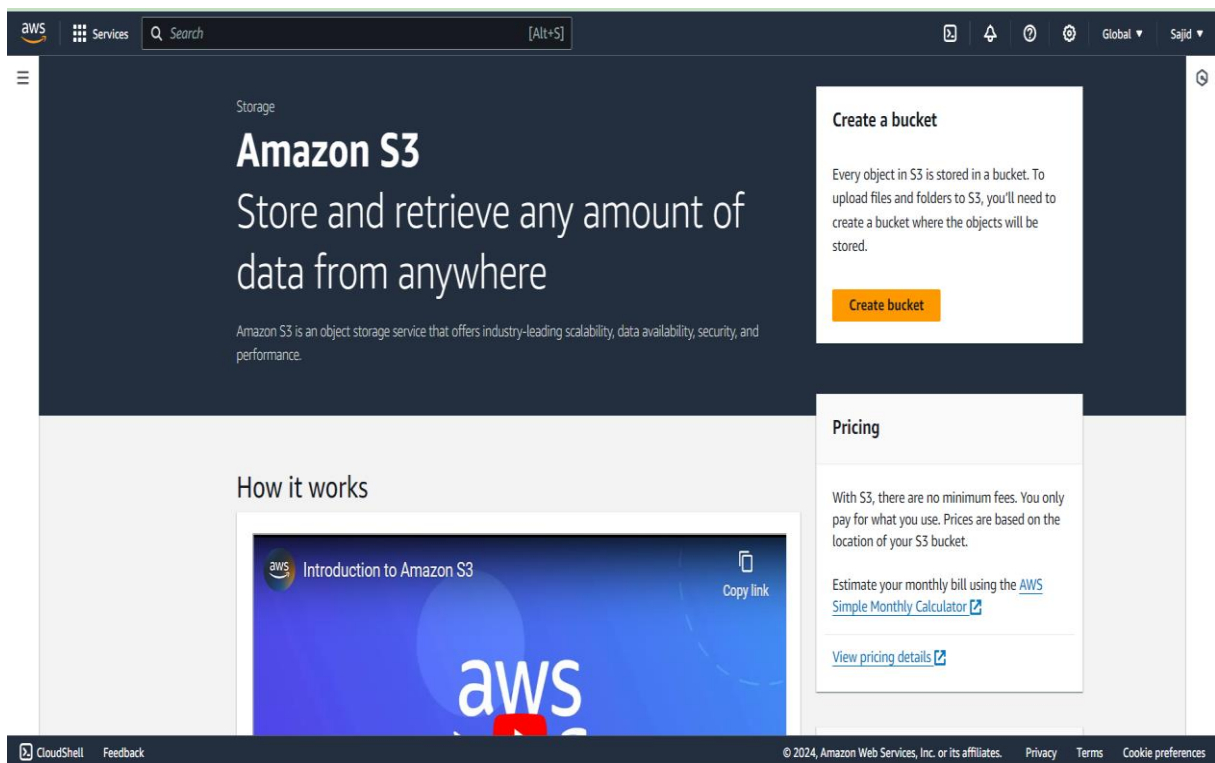


CONNECTING PYTHON Boto3 WITH AWS S3 BUCKET

-Sign in to the AWS Management console. After sign in, the screen appears is shown below:



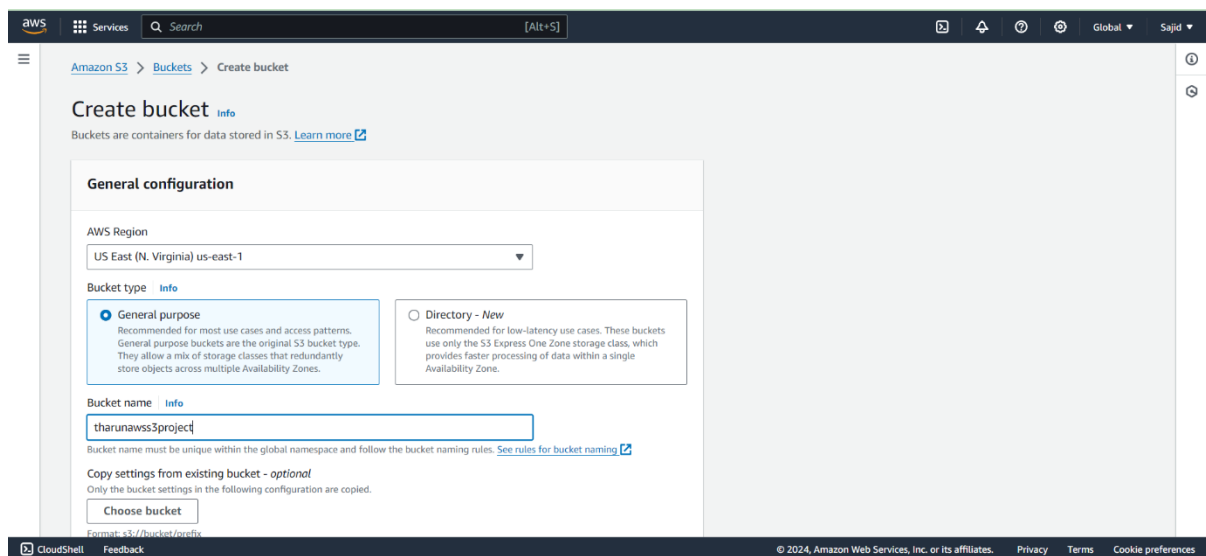
-Then move to the S3 services, which appears in console home. After clicking on S3, the screen appears is shown below:



-Next to create an S3 bucket, click on the "Create bucket". On clicking the "Create bucket" button, the screen appears is shown below:

-Enter the bucket name which should look like DNS address, and it should be resolvable. A bucket is like a folder that stores the objects. A bucket name should be unique. A bucket name should start with the lowercase letter, must not contain any invalid characters. It should be 3 to 63 characters long(tharunawss3project).

-Next, choose an AWS region nearest to your location or where you want your data to reside. In our case, it is (US East [N. Virginia]us-east-1).

The screenshot shows the AWS Management Console 'Create bucket' page. The breadcrumb navigation at the top reads 'Amazon S3 > Buckets > Create bucket'. The main heading is 'Create bucket' with an 'Info' link. Below this, a sub-header states 'Buckets are containers for data stored in S3. [Learn more](#)'. The 'General configuration' section contains three main fields: 'AWS Region' is a dropdown menu set to 'US East (N. Virginia) us-east-1'; 'Bucket type' has two radio button options: 'General purpose' (selected) and 'Directory - New'; 'Bucket name' is a text input field containing 'tharunawss3project'. Below the bucket name field, there is a note: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'. At the bottom of the configuration section, there is a link for 'Copy settings from existing bucket - optional' and a 'Choose bucket' button. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

-Keep default settings of ACL and Public Access.

-Next the Bucket Versioning category, choose Enabled. Bucket versioning is helpful when you want to track any changes in the file made, intentionally or unintentionally. You can see the previous versions of a file, retrieve it, restore it or preserve it.

-Leave other advance settings as default.

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**

-Click on the "Create" button. Now, the bucket is created.

Account snapshot [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets | Directory buckets

General purpose buckets (1) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access	Creation date
<input type="radio"/>	tharunawss3project	US East (N. Virginia) us-east-1	Bucket and objects not public	February 8, 2024, 20:07:51 (UTC+05:30)

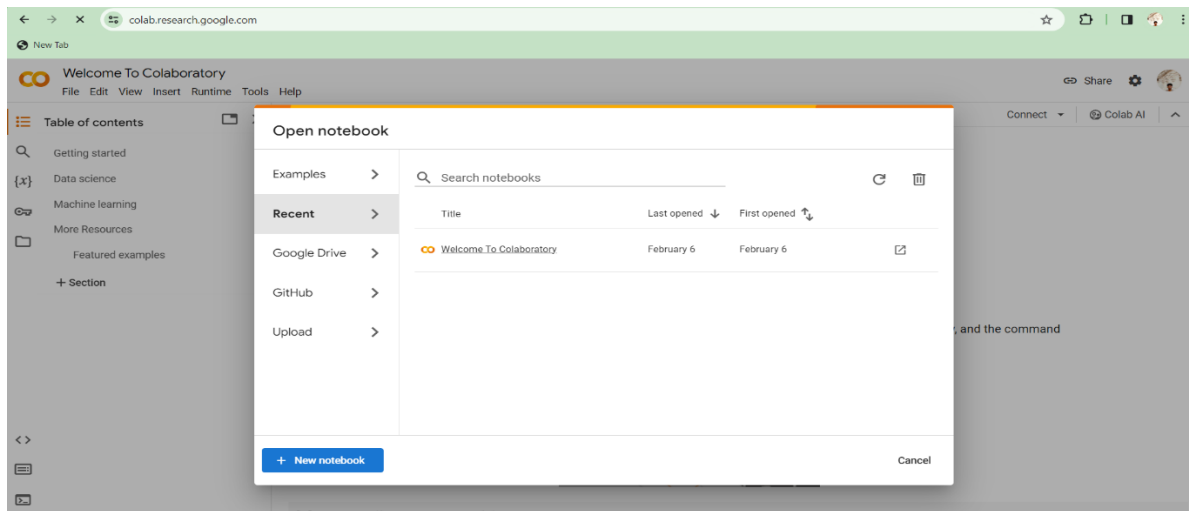
CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

-Now, click on the "tharunawss3project" and don't upload the files we can retrieve the data from python code.

-Then implementation of Boto3 in Python.

-Then now go to the google and open the GOOGLE COLAB to open Boto3.



-First, you need to install the boto3 module using pip, once the boto3 module is installed you can use it in your Python code.

-By using boto3 to list all the buckets in your AWS account.

-Boto3 is a powerful and versatile tool that can be used to automate, manage, and monitor various AWS resources and services.

```

pip install boto3

Collecting boto3
  Downloading boto3-1.16.49-py2.py3-none-any.whl (130 kB)
Collecting botocore<1.20.0,>=1.19.49
  Downloading botocore-1.19.49-py2.py3-none-any.whl (7.2 MB)
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting s3transfer<0.4.0,>=0.3.0
  Downloading s3transfer-0.3.3-py2.py3-none-any.whl (69 kB)
Requirement already satisfied: urllib3<1.27,>=1.25.4; python_version != "3.4" in c:\users\win10\anaconda3\envs\myenv\lib\site-packages (from botocore<1.20.0,>=1.19.49->boto3) (1.25.10)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\win10\anaconda3\envs\myenv\lib\site-packages (from botocore<1.20.0,>=1.19.49->boto3) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\win10\anaconda3\envs\myenv\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.20.0,>=1.19.49->boto3) (1.15.0)
Installing collected packages: jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.16.49 botocore-1.19.49 jmespath-0.10.0 s3transfer-0.3.3

[ ] import boto3
import pandas as pd

s3 = boto3.client('s3')

[ ] s3 = boto3.resource(
    service_name='s3',
    region_name='us-east-2',
    aws_access_key_id='',
    aws_secret_access_key=''
  )
  
```

-Then we first create a boto3 session by providing our AWS access key ID, secret access key, and AWS region.

-Then create an s3 client using the session and use the list buckets method to get a list of all the buckets in our account.

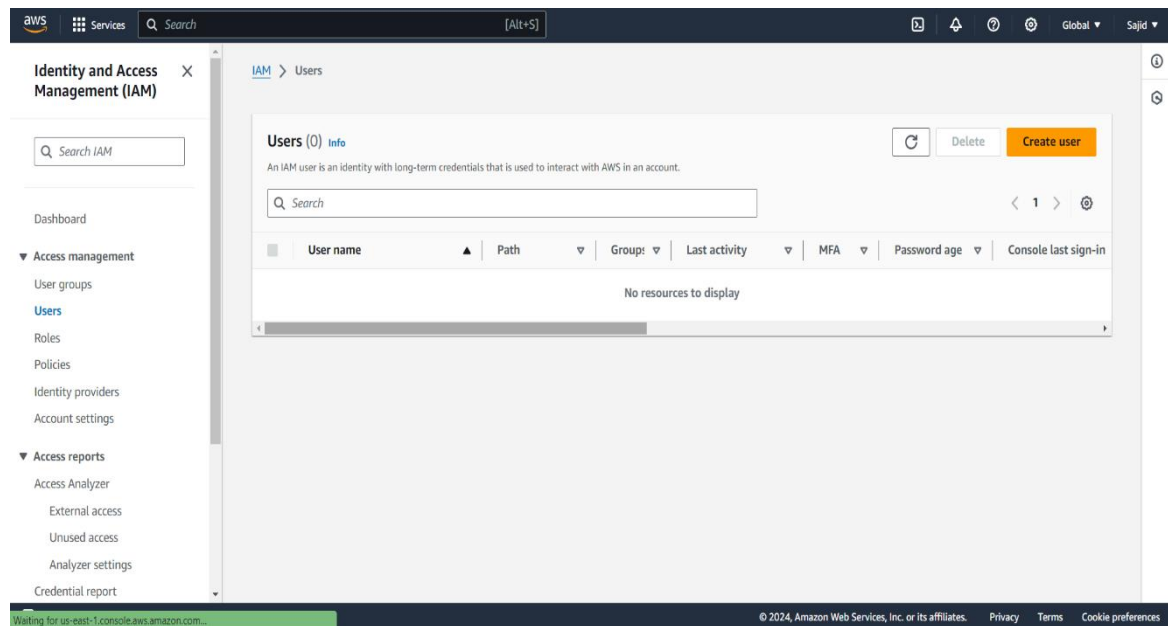
-After that iterate through the list of buckets and print their names.

-Note that you need to replace your_access_key_id, your_secret_access_key, and your aws_region with your actual AWS access key ID, secret access key, and AWS region.

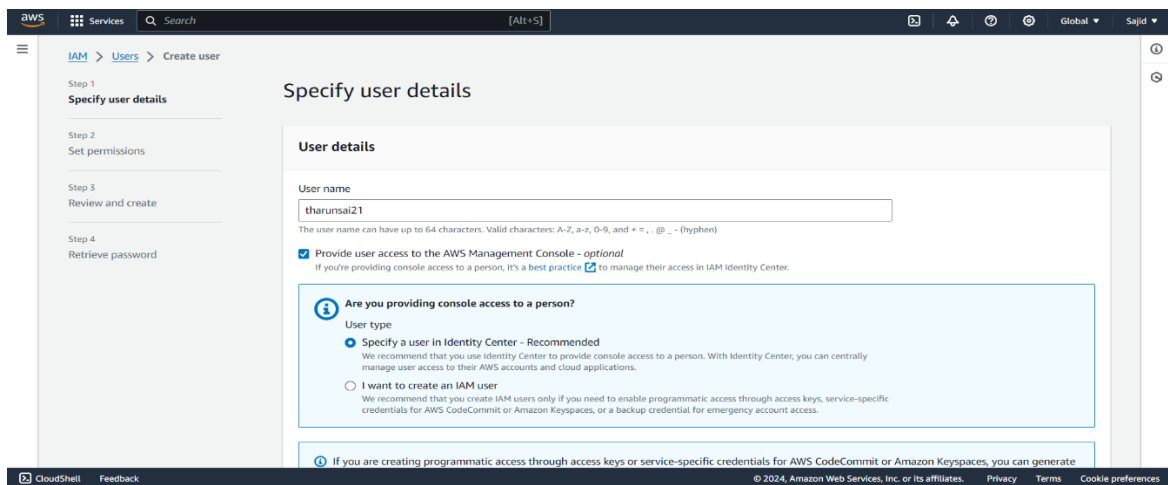
-And also make sure that the IAM user associated with the access key has the necessary permissions to access S3.

- For creation AWS access key ID, secret access key go to the AWS account and open the IAM user.

-Click on Create user.



-Then enter the user name(tharunsai21).



-Then select **I want to create an IAM user**.

-Then select **Custom password** and enter the password you want to create.

-After that click on “Next”.

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

☒ Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

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

-After that Set the Permissions by selecting the Attach policies directly.

Permissions policies (1/1186)
Choose one or more policies to attach to your new user.

Filter by Type: All types 15 matches

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonDMSRedshiftS3Role	AWS managed	0
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	0
<input type="checkbox"/> AmazonS3ObjectLambdaExecution...	AWS managed	0
<input type="checkbox"/> AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/> AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3...	AWS managed	0
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3...	AWS managed	0
<input type="checkbox"/> AWSS3OnOutpostsServiceRolePolicy	AWS managed	0
<input type="checkbox"/> IVSRecordToS3	AWS managed	0
<input type="checkbox"/> QuickSightAccessForS3StorageMan...	AWS managed	0

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

-And then search the policy to give the Full access of S3(AmazonS3FullAccess).

-After giving the policy click on “Next”.

<input type="checkbox"/> AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/> AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3...	AWS managed	0
<input type="checkbox"/> AWSBackupServiceRolePolicyForS3...	AWS managed	0
<input type="checkbox"/> AWSS3OnOutpostsServiceRolePolicy	AWS managed	0
<input type="checkbox"/> IVSRecordToS3	AWS managed	0
<input type="checkbox"/> QuickSightAccessForS3StorageMan...	AWS managed	0
<input type="checkbox"/> s3scr_for_firstbucket2031_14a97f	Customer managed	0
<input type="checkbox"/> s3scr_for_secondbucket392001_b9a288	Customer managed	0
<input type="checkbox"/> s3replcate_for_secondbucket392001_...	Customer managed	0
<input type="checkbox"/> S3StorageLensServiceRolePolicy	AWS managed	0

► Set permissions boundary - optional

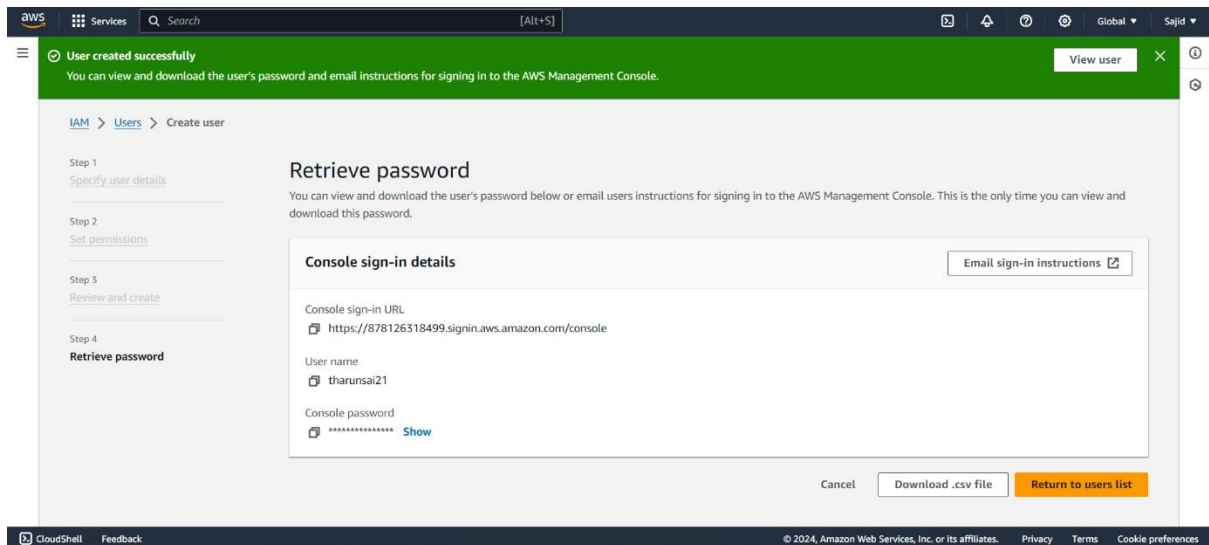
Cancel Previous Next

CloudShell Feedback

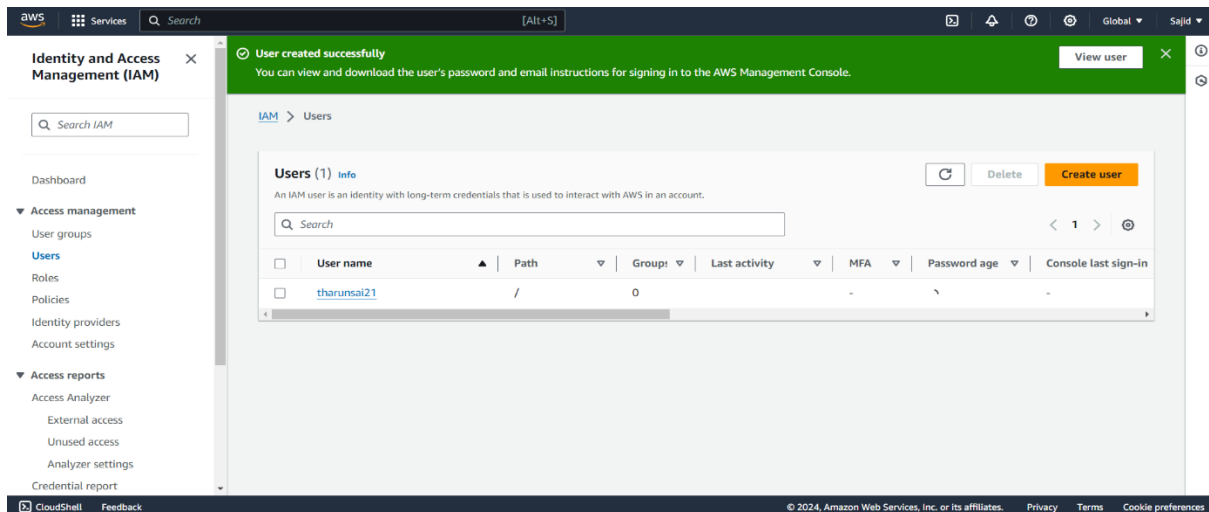
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

-After that you will get the details of user and policy you gave.

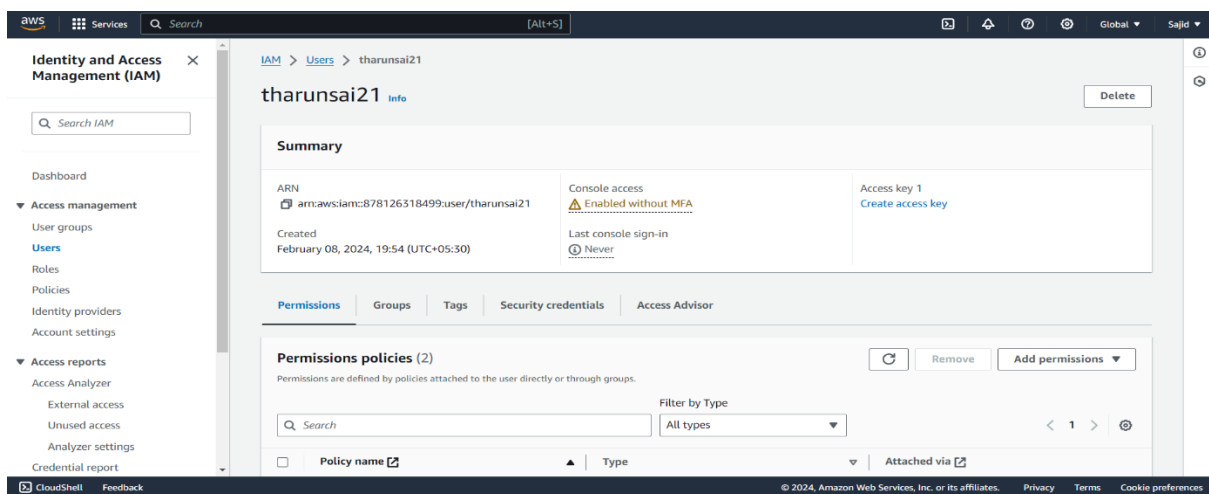
-Console sign-in details will display and click on Return to users list.



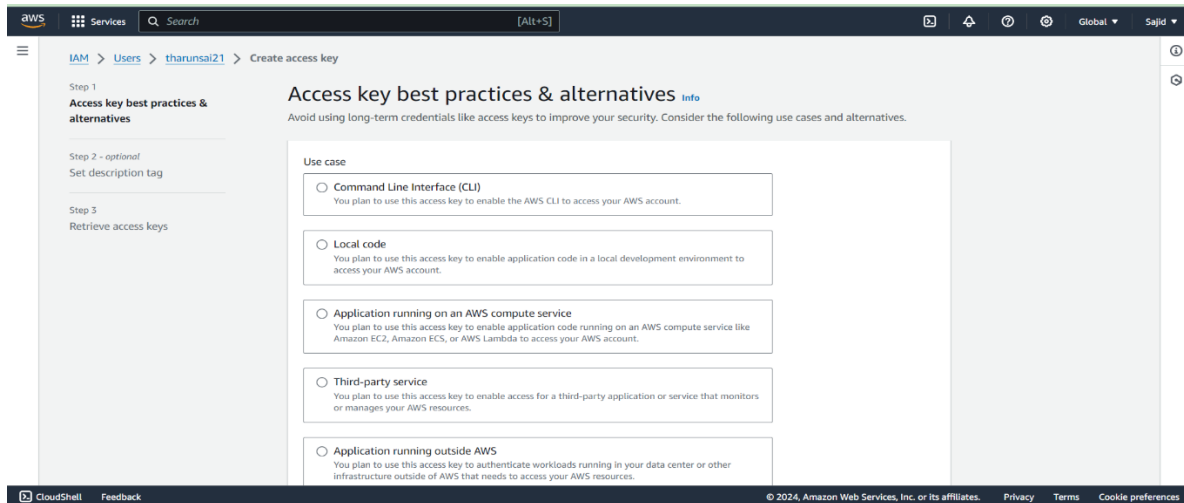
-Then Click on user name “tharunsai21”.



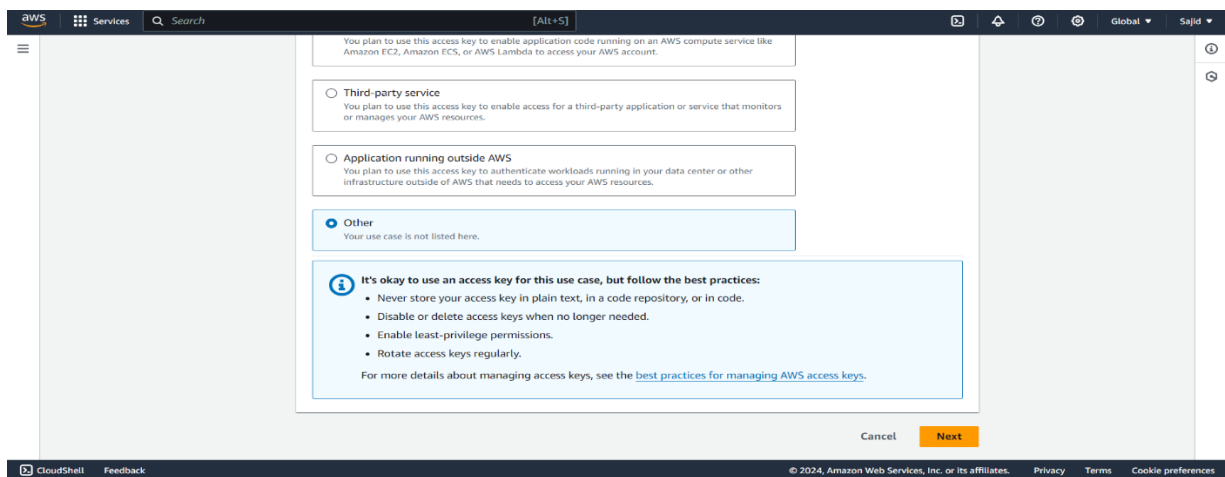
-It displays summary of users and click on “Create access key”.



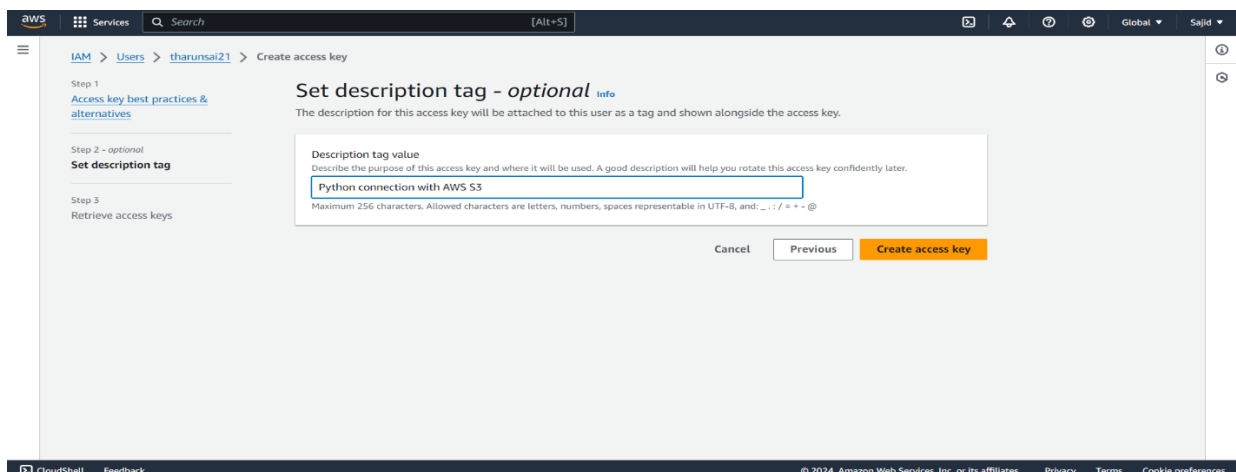
-After click on create select the Access key best practices & alternatives.



-Select Other and Click on Next button.

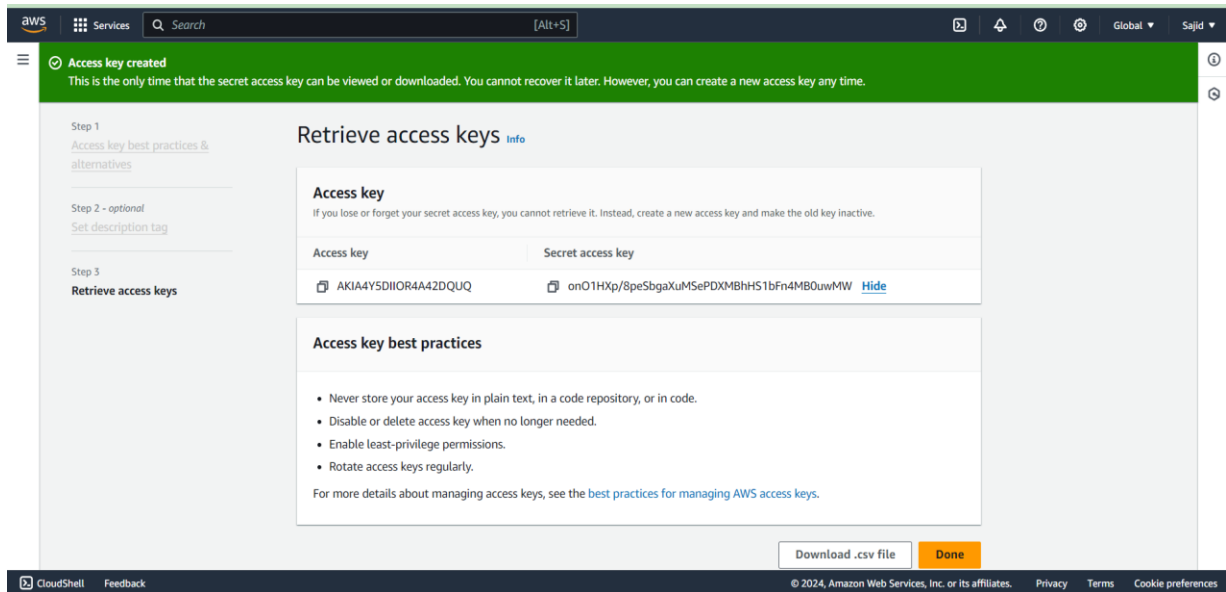


-After that Give the Description tag value as your wish(optional) “python connection with AWS S3”. Then click on Create access key.



-After that you will get the Access key and Secret Access key.

-Access key (AKIA4Y5DIIOR4A42DQUQ), Secret access key(on01HXp/8peSbgaXuMSePDXMB0uwMW).



-Apply this Access key (AKIA4Y5DIIOR4A42DQUQ), Secret access key(on01HXp/8peSbgaXuMSePDXMB0uwMW) and bucket name in the AWS boto3 Python.

```
boto3 read S3.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[ ] import os
os.environ["AWS_DEFAULT_REGION"] = "us-east-1"
os.environ["AWS_ACCESS_KEY_ID"] = "AKIA4Y5DIIOR4A42DQUQ"
os.environ["AWS_SECRET_ACCESS_KEY"] = "on01HXp/8peSbgaXuMSePDXMBhS1bFn4MB0uwMW"

import pandas as pd

[ ] import pandas as pd

# Make dataframes
foo = pd.DataFrame({'x': [1, 2, 3], 'y': ['a', 'b', 'c']})
bar = pd.DataFrame({'A': [10, 20, 30], 'B': ['aa', 'bb', 'cc']})

# Save to csv
foo.to_csv('foo.csv')
bar.to_csv('bar.csv')

[ ] # Upload files to S3 bucket
s3.Bucket('tharunawss3project').upload_file(filename='foo.csv', Key='foo.csv')
s3.Bucket('tharunawss3project').upload_file(filename='bar.csv', Key='bar.csv')

[ ] for obj in s3.Bucket('tharunawss3project').objects.all():
    print(obj)

0s completed at 8:03 PM
```

-Upload a files to the S3 bucket “tharunawss3project” using the following code.

```
boto3 read S3.ipynb
File Edit View Insert Runtime Tools Help Saving failed since 8:15 PM

+ Code + Text

[10] # Upload files to S3 bucket
s3.Bucket('tharunawss3project').upload_file(filename='foo.csv', Key='foo.csv')
s3.Bucket('tharunawss3project').upload_file(filename='bar.csv', Key='bar.csv')

[11] for obj in s3.Bucket('tharunawss3project').objects.all():
    print(obj)

s3.ObjectSummary(bucket_name='tharunawss3project', key='bar.csv')
s3.ObjectSummary(bucket_name='tharunawss3project', key='foo.csv')

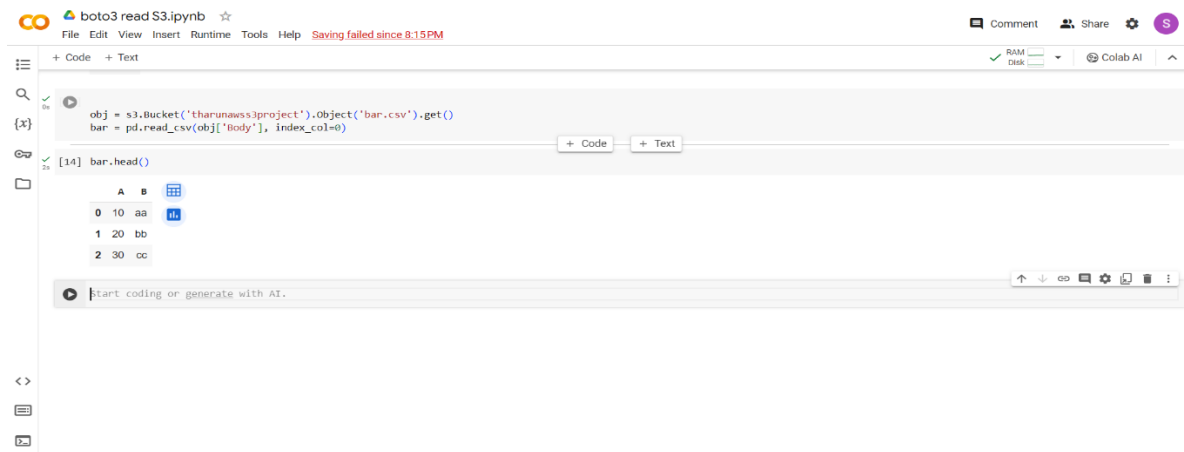
# Load csv file directly into python
obj = s3.Bucket('tharunawss3project').Object('foo.csv').get()
foo = pd.read_csv(obj['Body'], index_col=0)

foo.head()

   x  y
0  1  a
1  2  b
2  3  c

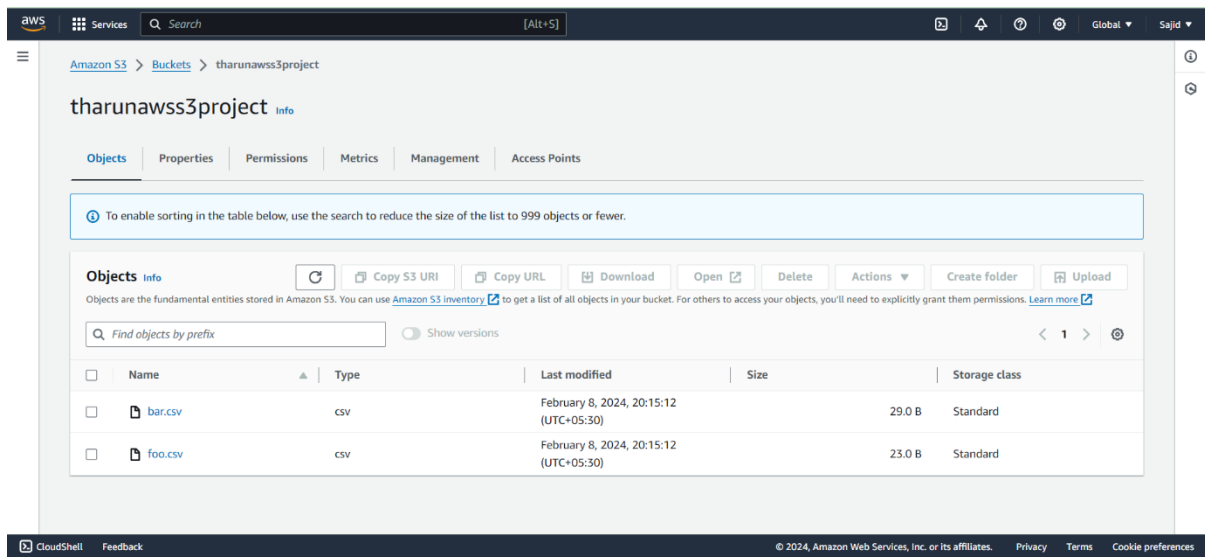
Automatic saving failed. This file was updated remotely or in another tab. Show diff
0s completed at 8:15 PM
```

-Files of bar and foo will be write as a code.



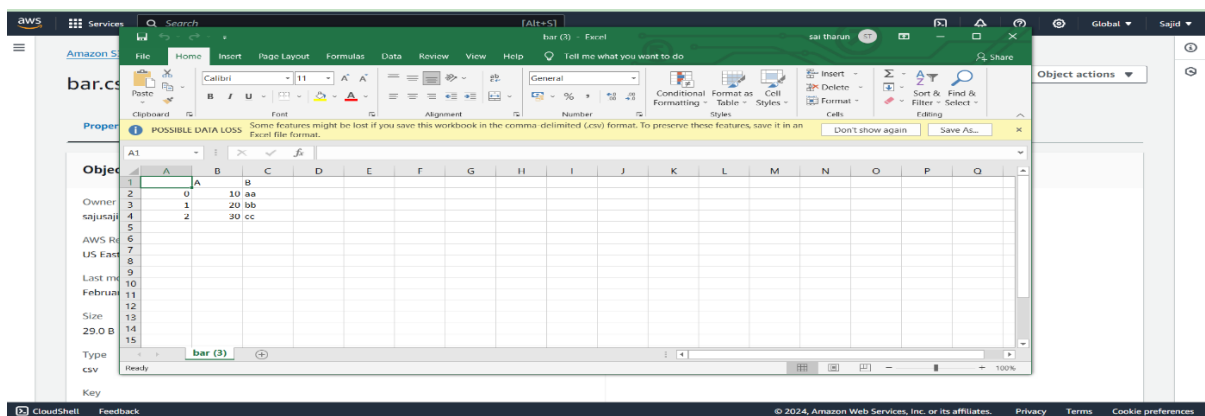
-Run all the code and open the S3 bucket.

-Then the files of bar and foo automatically upload in a bucket after running the code.

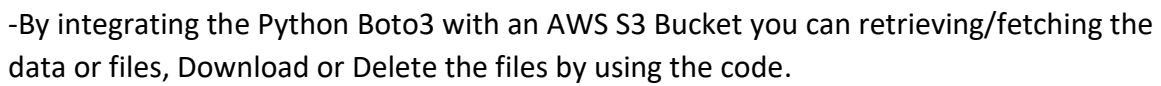


-Download and delete the files to the s3 bucket using the following code.

-Download the bar file by clicking on the **bar.csv**.



-Download the bar file by clicking on the **foo.csv**.



V SAI THARUN GOUD