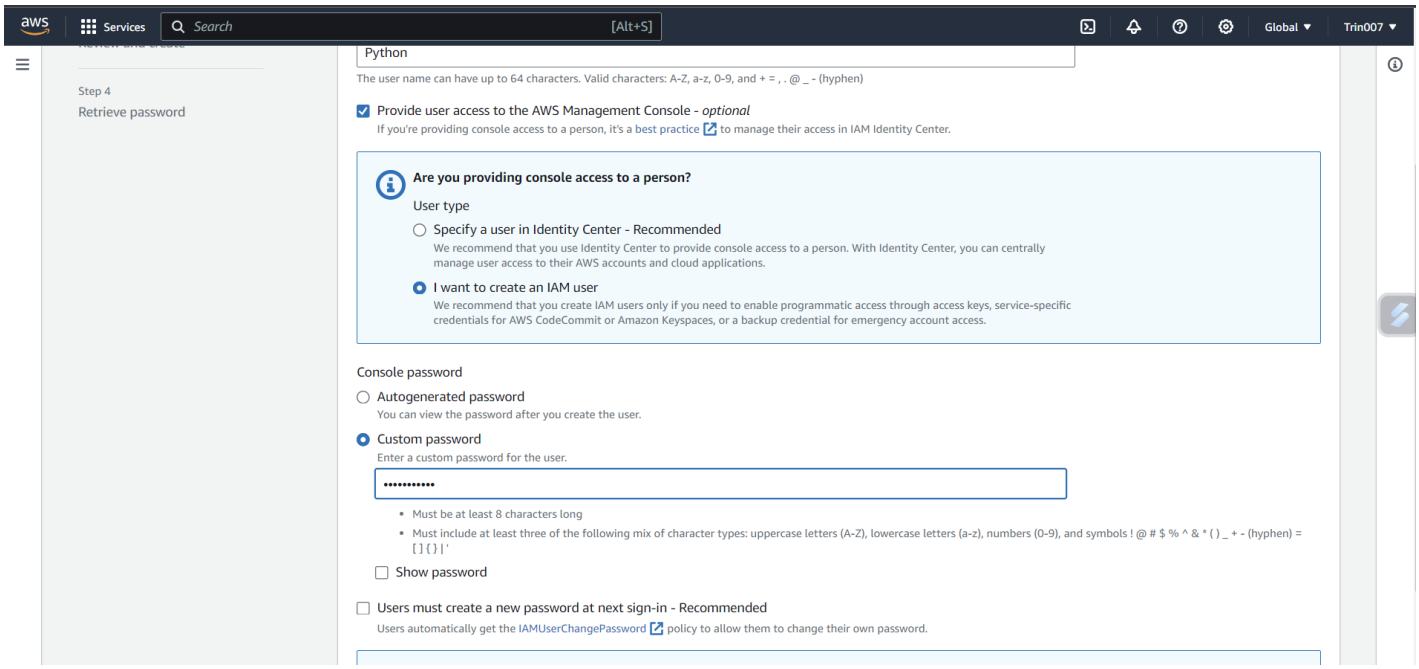
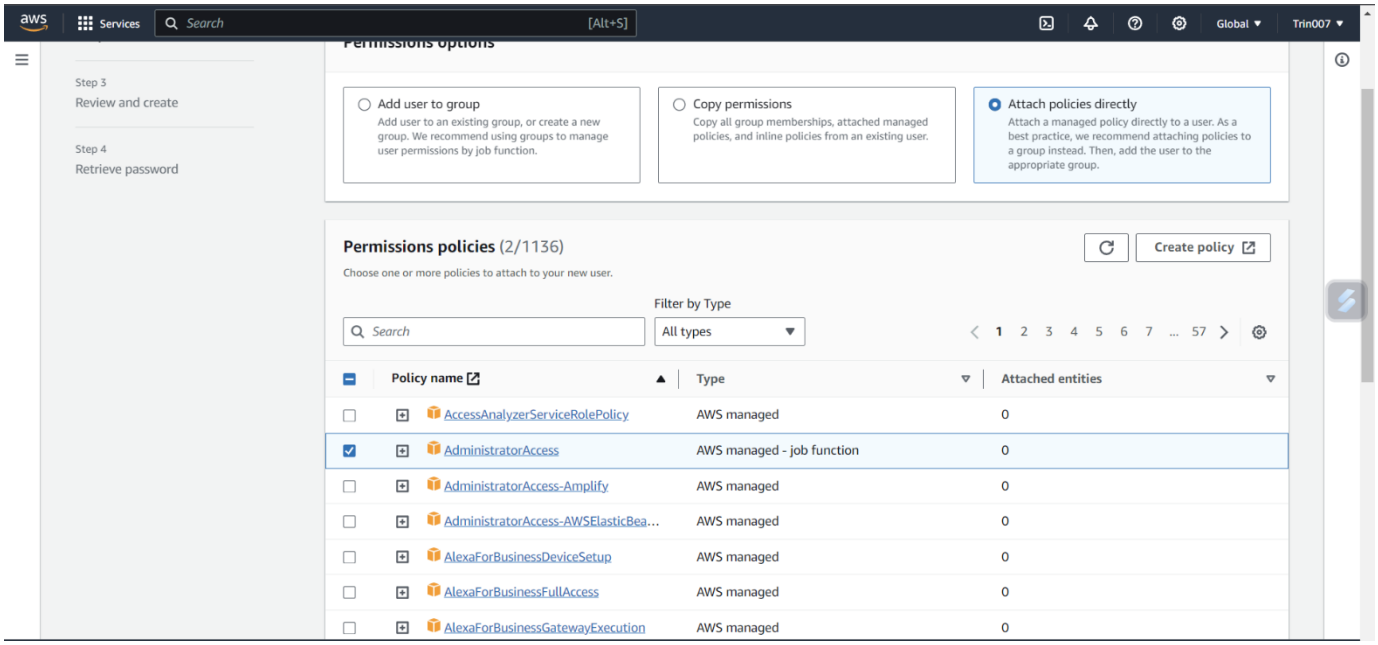
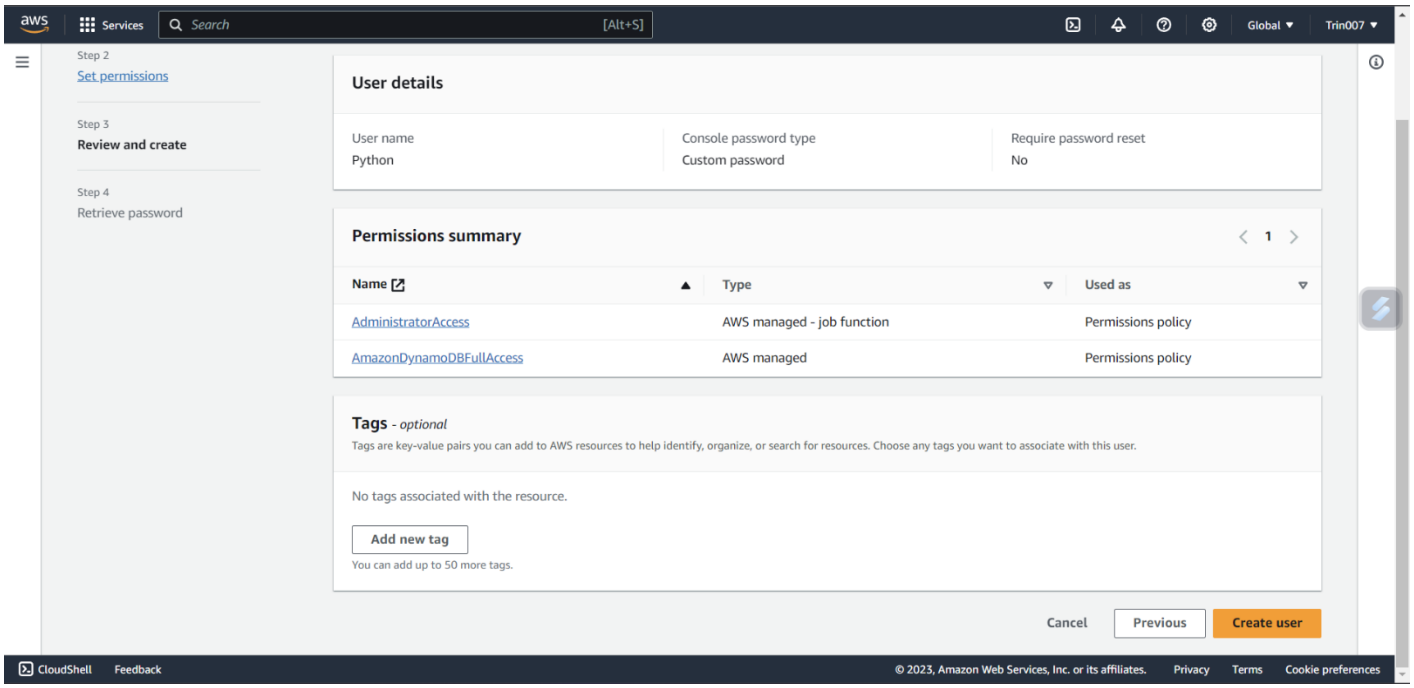
ACCESSING DYNAMOD OF AWS SERVICE USING THIRD-PARTY SERVICE

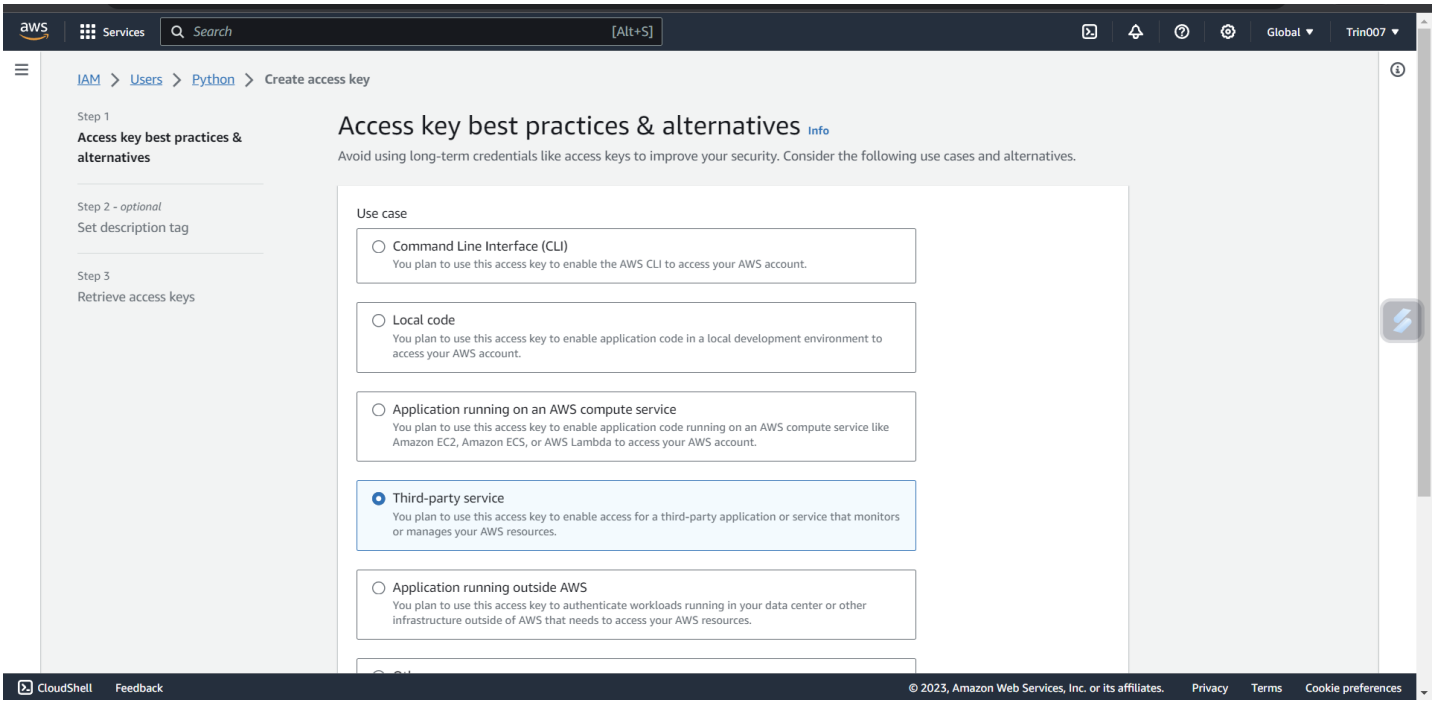
- First create an Amazon AWS root user: https://shorturl.at/dIJOW
- Then navigate to IAM from the search bar
- Inside IAM select the **users** from the left navbar
- Click Add New User and create an IAM user with two policies:
    - DynamoDB full access
    - Administrator access



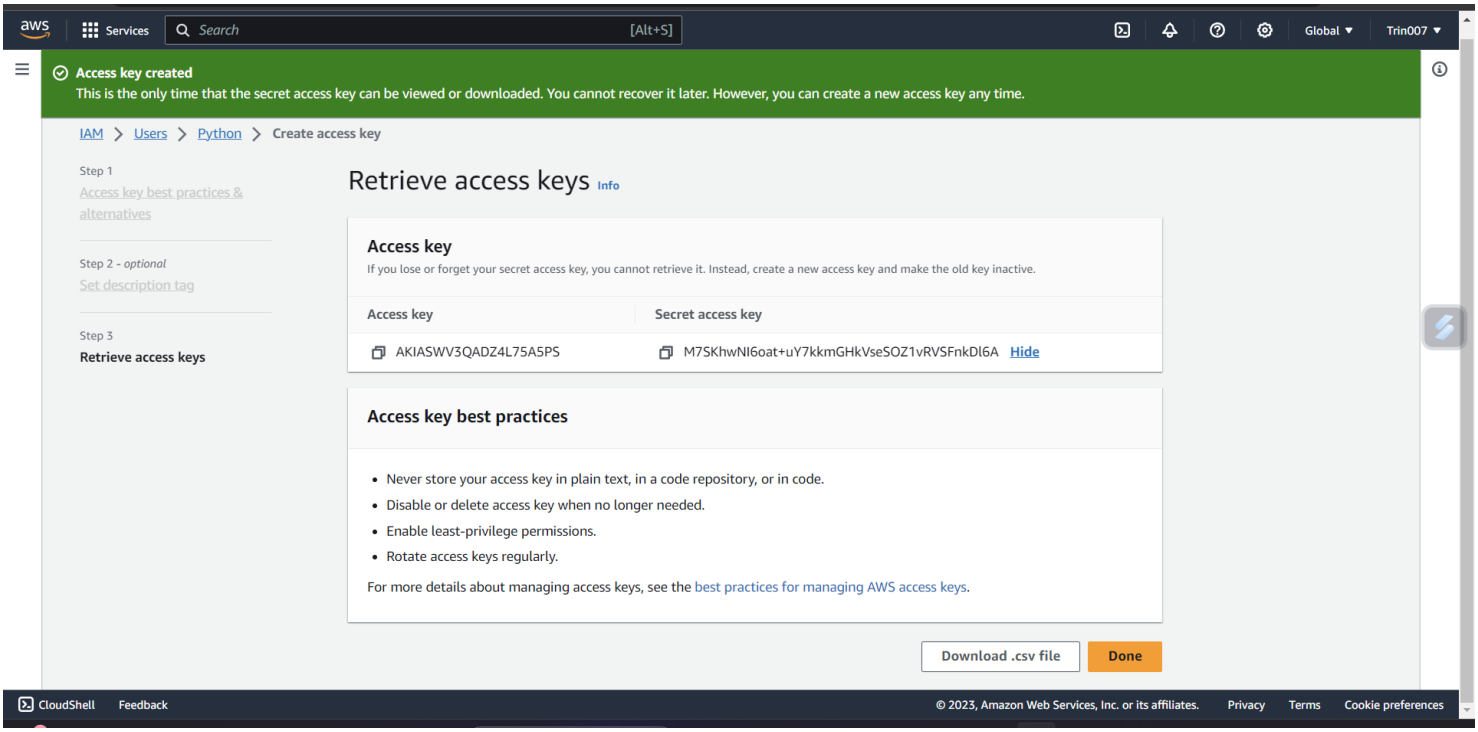




- Then return to the user's page and select the newly created user
- Inside that click create access key and click on "**Third Party access**".



- Give a tag (optional) and create an access key. Note down the Access Key and Secret Access Key before returning to the user's page.



- Now go to search and type “**DynamoDB**” and click on that
- Then on the top right near your username set the region as Mumbai(ap-south-1)
- Now create a Python file and write the program with the access key, secret access key, and region-name.

Python Program

```
import boto3
Access_Key="AKIASWV3QADZRAVW2XOK"
Secret_Access_Key="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"
client = boto3.client(
   'dynamodb',
   aws_access_key_id=Access_Key,
   aws_secret_access_key=Secret_Access_Key,
   region_name="ap-south-1",
   )
dynamodb = boto3.resource(
   'dynamodb',
   aws_access_key_id=Access_Key,
   aws_secret_access_key=Secret_Access_Key,
   region_name="ap-south-1",
   )
ddb_exceptions = client.exceptions
try:
   table = client.create_table(
```

```
        TableName='ISS_locations',
        KeySchema=[
            {
                'AttributeName': 'timestamp',
                'KeyType': 'HASH'
            }
        ],
        AttributeDefinitions=[
            {
                'AttributeName': 'timestamp',
                'AttributeType': 'N'
            }

        ],
        ProvisionedThroughput={
            'ReadCapacityUnits': 10,
            'WriteCapacityUnits': 10
        }
    )
    print("Creating table")
    waiter = client.get_waiter('table_exists')
    waiter.wait(TableName='ISS_locations')
    print("Table created")

except ddb_exceptions.ResourceInUseException:
    print("Table exists")

import requests
import time
def call_ISS_API():
    print("Calling ISS")
    time.sleep(1)
    r = requests.get('http://api.open-notify.org/iss-now.json')
    json = r.json()
    json['latitude'] = json['iss_position']['latitude']
    json['longitude'] = json['iss_position']['longitude']
    del json['iss_position']
    del json['message']
    return json

api_calls = {}
for i in range(5):
    response = call_ISS_API()
    api_calls[response['timestamp']] = response

print("Putting items")
```

```
for response in api_calls:
   dynamodb.Table('ISS_locations').put_item(
      Item=api_calls[response]
      )

print("Scanning table")
response = dynamodb.Table('ISS_locations').scan()
for i in response['Items']:
   print(i)
```

Replace the Access_Key and Secret_Access_key with your own in the above program.
This program will create a table called “ISS_Location” and add 5 items in it.