CS 551: Assignment 6

Ishaan Roychowdhury

Function handler code for part 4:

```
import json
import urllib.parse
import boto3
s3 = boto3.client('s3')
def fibo_homework6(n):
    fib_table_ishaanHW = [0] * (n + 1)
    fib_table_ishaanHW[0] = 0
    if n > 0:
        fib_table_ishaanHW[1] = 1
    for i in range(2, n + 1):
        fib_table_ishaanHW[i] = fib_table_ishaanHW[i-1] + fib_table_ishaanHW[i-2]
    return fib_table_ishaanHW[n]

def lambda_handler(event, context):
    if 'Records' not in event or not event['Records']:
        print("No records")
        return "No records"
    fetch_Rec = event['Records'][0]
    if 's3' not in fetch_Rec or 'bucket' not in fetch_Rec['s3'] or 'name' not in
fetch_Rec['s3']['bucket']:
        print("Invalid S3 scene bro")
        return "Invalid S3 scene"
    bucket = fetch_Rec['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(fetch_Rec['s3']['object']['key'], encoding='utf-
8')
    get_response = s3.get_object(Bucket=bucket, Key=key)
    if 'Body' not in get_response:
        print(f"Missing 'Body' in S3 object for {key}")
        return f"Missing data in S3 object {key}"
    fetch_json = json.loads(get_response['Body'].read().decode('utf-8'))
    if 'n' not in fetch_json:
        print("JSON does not contain 'n'")
        return "Invalid JSON content"
    n = int(fetch_json['n'])
    retVal_for_Fib = fibo_homework6(n)
    final_json_to_upload_to_S3 = {'n': n, 'fibonacci_result': retVal_for_Fib}
    key_for_homework = f"fibonacci_result_{n}.json"
    s3.put_object(Bucket=bucket, Key=key_for_homework,
Body=json.dumps(final_json_to_upload_to_S3))
    print(f"I have computed fibonacci for this homework and the results for n={n} is:
{retVal_for_Fib}")
    return f"I have computed the results of fibonacci in the assignment and those
results for n={n} have been uploaded to S3: {key_for_homework}"
```

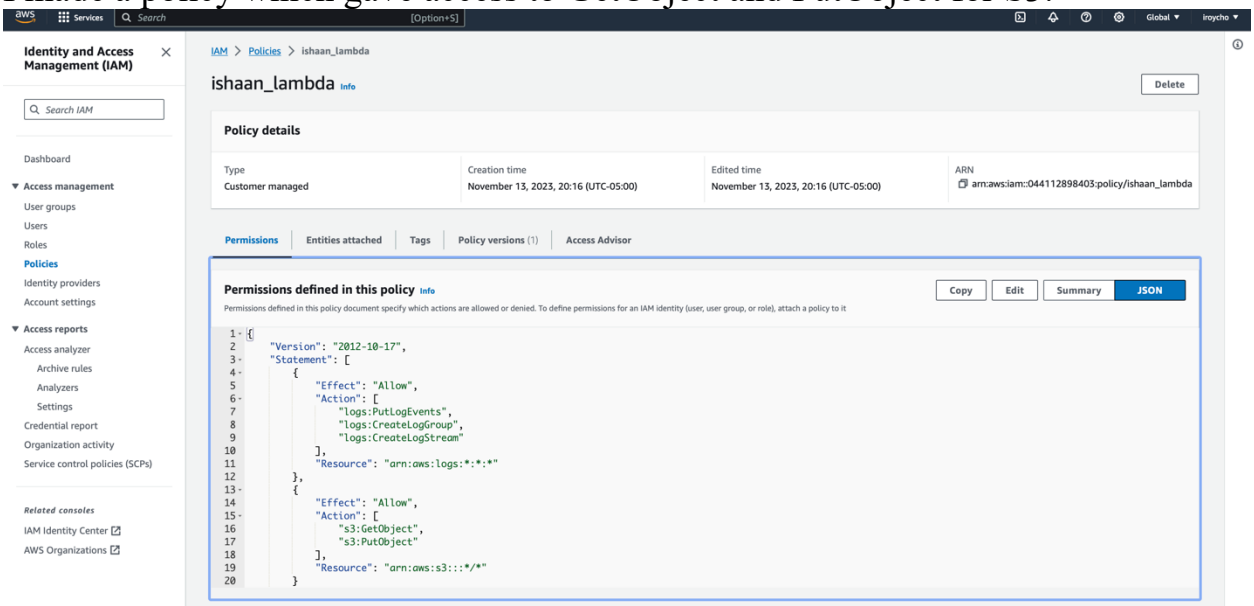
This was my code for function handler in part 4:

app.py > lambda_handler

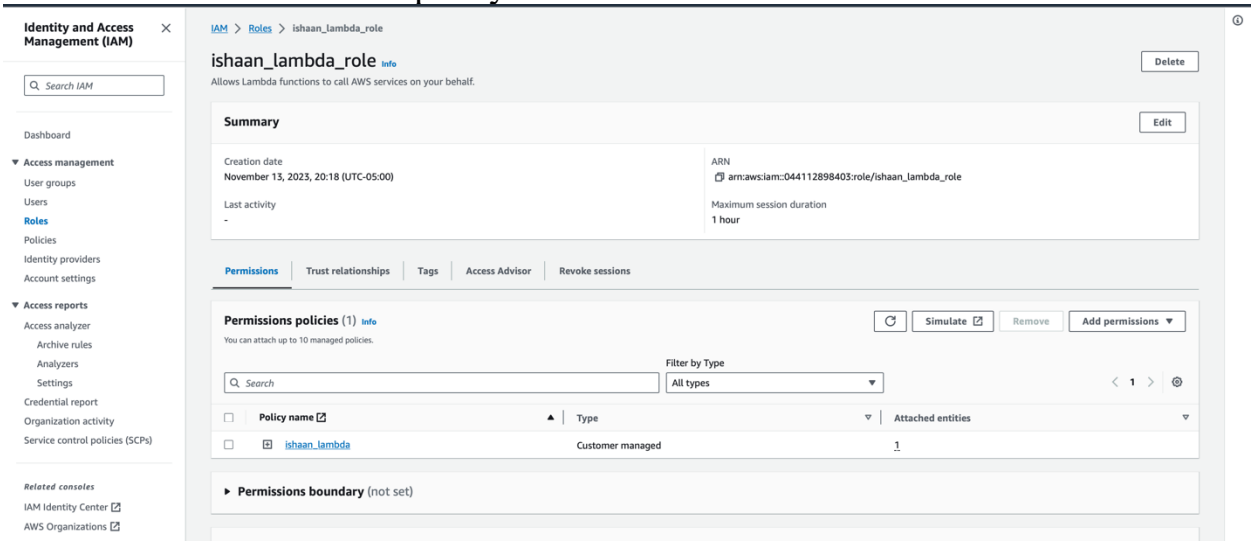
```python
import json
import urllib.parse
import boto3
s3 = boto3.client('s3')
def fibo_homework6(n):
    fib_table_ishaanHW = [0] * (n + 1)
    fib_table_ishaanHW[0] = 0
    if n > 0:
        fib_table_ishaanHW[1] = 1
    for i in range(2, n + 1):
        fib_table_ishaanHW[i] = fib_table_ishaanHW[i-1] + fib_table_ishaanHW[i-2]
    return fib_table_ishaanHW[n]

def lambda_handler(event, context):
    if 'Records' not in event or not event['Records']:
        print("No records")
        return "No records"
    fetch_Rec = event['Records'][0]
    if 's3' not in fetch_Rec or 'bucket' not in fetch_Rec['s3'] or 'name' not in fetch_Rec['s3']['bucket']:
        print("Invalid S3 scene bro")
        return "Invalid S3 scene"
    bucket = fetch_Rec['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(fetch_Rec['s3']['object']['key'], encoding='utf-8')
    get_response = s3.get_object(Bucket=bucket, Key=key)
    if 'Body' not in get_response:
        print(f"Missing 'Body' in S3 object for {key}")
        return f"Missing data in S3 object {key}"
    fetch_json = json.loads(get_response['Body'].read().decode('utf-8'))
    if 'n' not in fetch_json:
        print("JSON does not contain 'n'")
        return "Invalid JSON content"
    n = int(fetch_json['n'])
    retVal_for_Fib = fibo_homework6(n)
    final_json_to_upload_to_S3 = {'n': n, 'fibonacci_result': retVal_for_Fib}
    key_for_homework = f"fibonacci_result_{n}.json"
    s3.put_object(Bucket=bucket, Key=key_for_homework, Body=json.dumps(final_json_to_upload_to_S3))
    print(f"I have computed fibonacci for this homework and the results for n={n} is: {retVal_for_Fib}")
    return f"I have computed the results of fibonacci in the assignment and those results for n={n} have been uploaded to S3: {key_for_homework}"
```

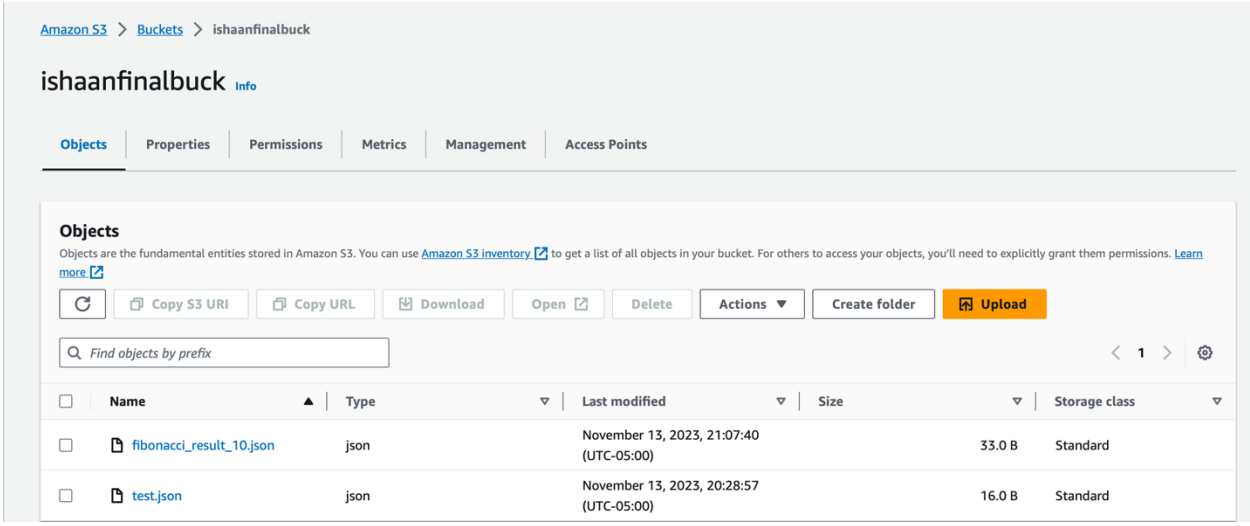
I made a policy which gave access to GetObject and PutObject for S3:



I made a role that used this policy:

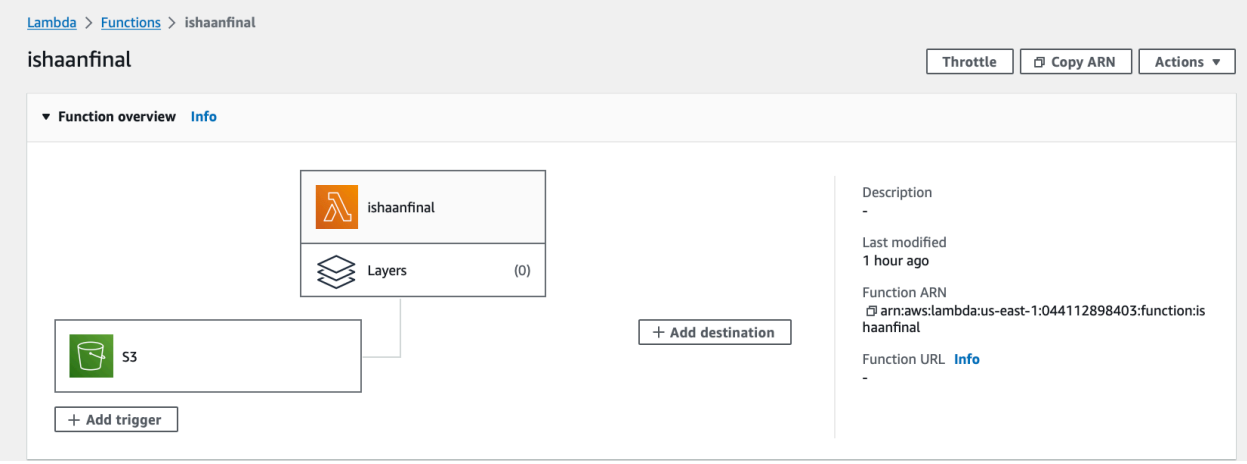


I made a S3 bucket for uploading json files, both from my end and from lambda's end. I used the same bucket to write the JSON result:

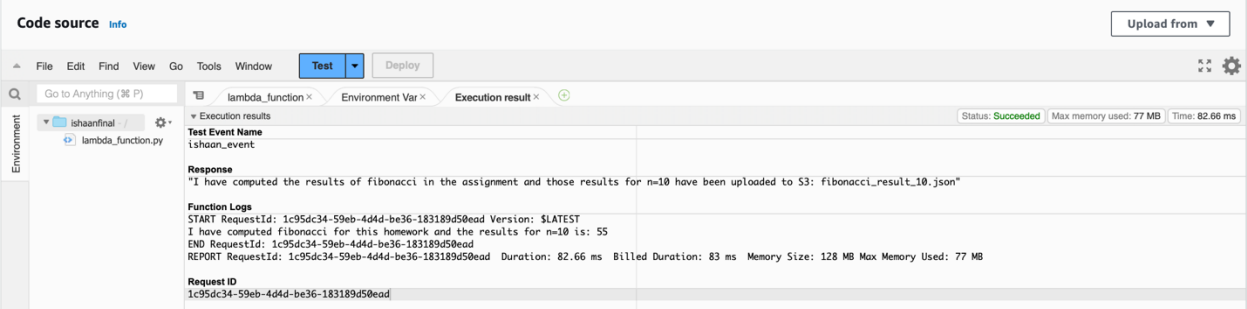


(In this I uploaded, test.json and once I ran the lambda function, fibonacci_result_10.json got added to the same bucket with the result)

I set up a trigger for S3 and my bucket in Lambda:

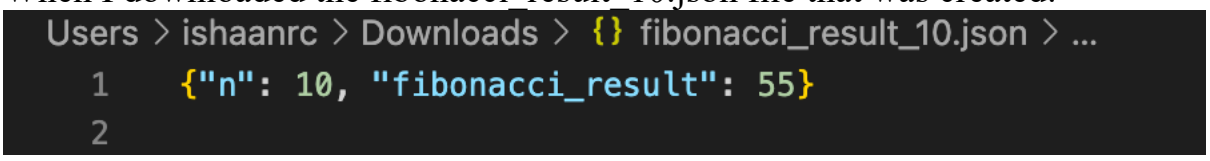


Finally, I ran a test to see if my code works and it does!!:



test.json that I uploaded:

When I downloaded the fibonacci_result_10.json file that was created:



This successfully shows me the 10th number in the Fibonacci Series which is 55. Hence this works!! ☺