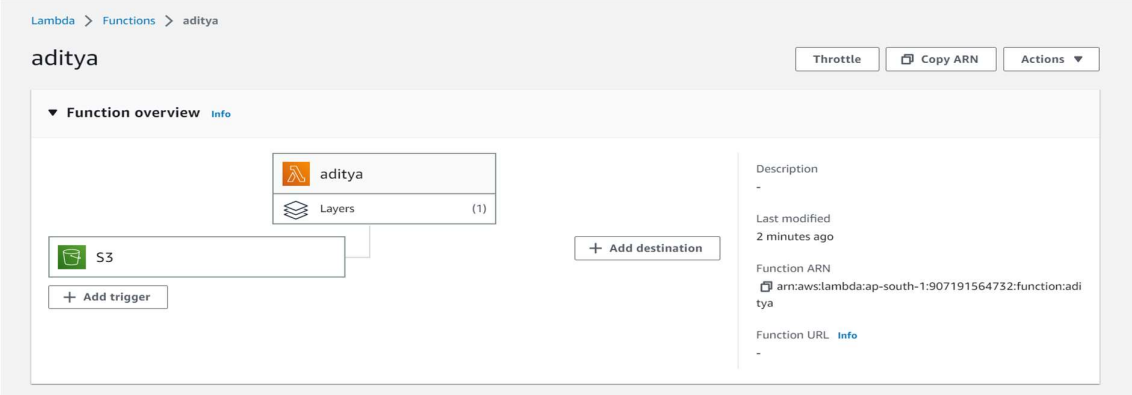
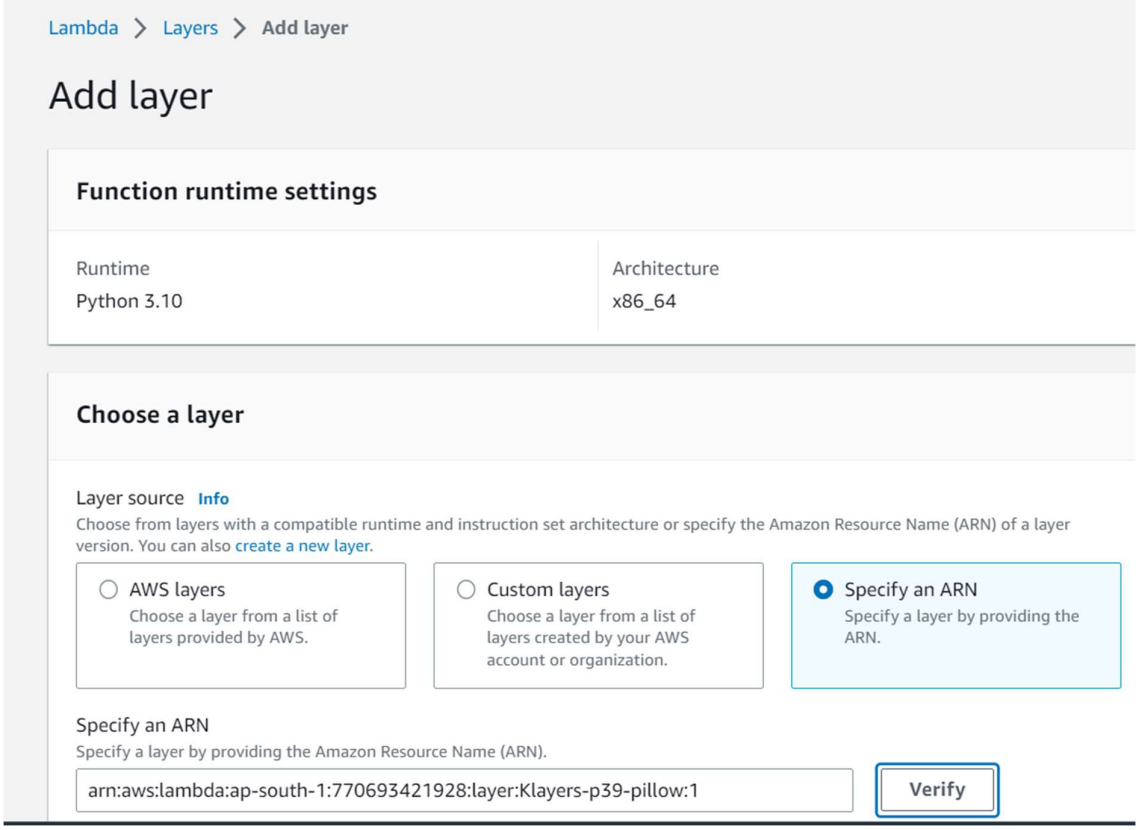
Aditya Kumar Week 2 Task

Task 1: Compression of an Image file using Lambda and S3 using S3 trigger.

Step 1: Created a lambda function and added s3 as trigger and also added a pillow library layer.





Description

pillow==9.2.0 | 9ed49f6351846301b8a50b0fc2c4196785e5577424d08534e402bb7c40727e74

Compatible runtimes

Python 3.6, Python 3.7, Python 3.8, Python 3.9

Compatible architectures

-

Cancel Add

Step 2: Write the python code to compress the image.

Code source Info

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

aditya

lambda_function.py

lambda_function × Execution results ×

```python
import boto3
from io import BytesIO
from PIL import Image

s3 = boto3.client('s3')

def lambda_handler(event, context):
    # Get the bucket and object key from the S3 event
    bucket_name = 'week2-aditya'
    object_key = 'image/abc.jpg'

    # Read the object content from S3
    response = s3.get_object(Bucket=bucket_name, Key=object_key)
    image_content = response['Body'].read()

    # Compress the image
    img = Image.open(BytesIO(image_content))
    compressed_buffer = BytesIO()
    img.save(compressed_buffer, format='JPEG', quality=50)
    compressed_content = compressed_buffer.getvalue()

    # Store the compressed image in another S3 bucket
    target_bucket_name = 'week2-aditya'
    target_object_key =  object_key
    s3.put_object(Bucket=target_bucket_name, Key=target_object_key, Body=compressed_content)

    # Log the success message
    print(f'Successfully stored compressed image {object_key} in S3 bucket {target_bucket_name}.')

    return {
        'statusCode': 200,
        'body': 'Compressed image stored successfully in S3'
    }
```

Step 3: Verifying the image size before Testing the code.

Amazon S3 > Buckets > week2-aditya > image/

image/

Copy S3 URI

Objects Properties

Objects (1)

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

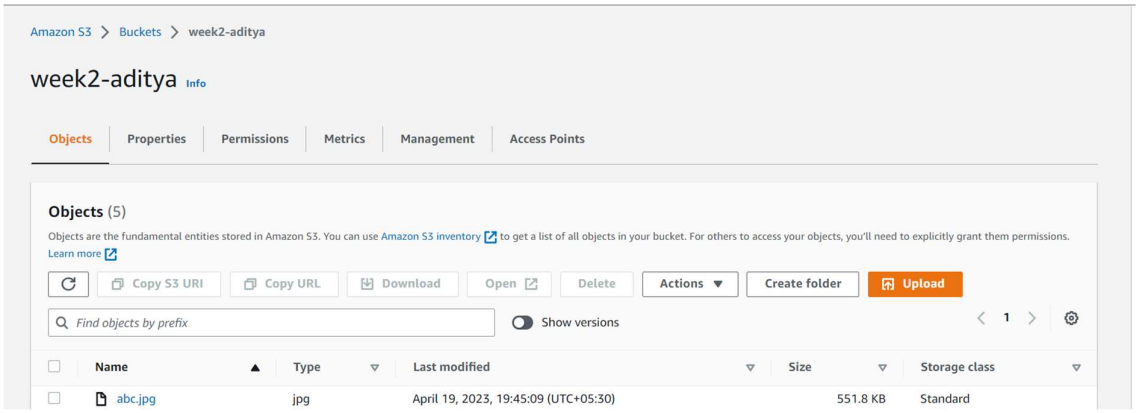
Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix Show versions 1

Name	Type	Last modified	Size	Storage class
abc.jpg	jpg	April 19, 2023, 19:29:45 (UTC+05:30)	1.1 MB	Standard

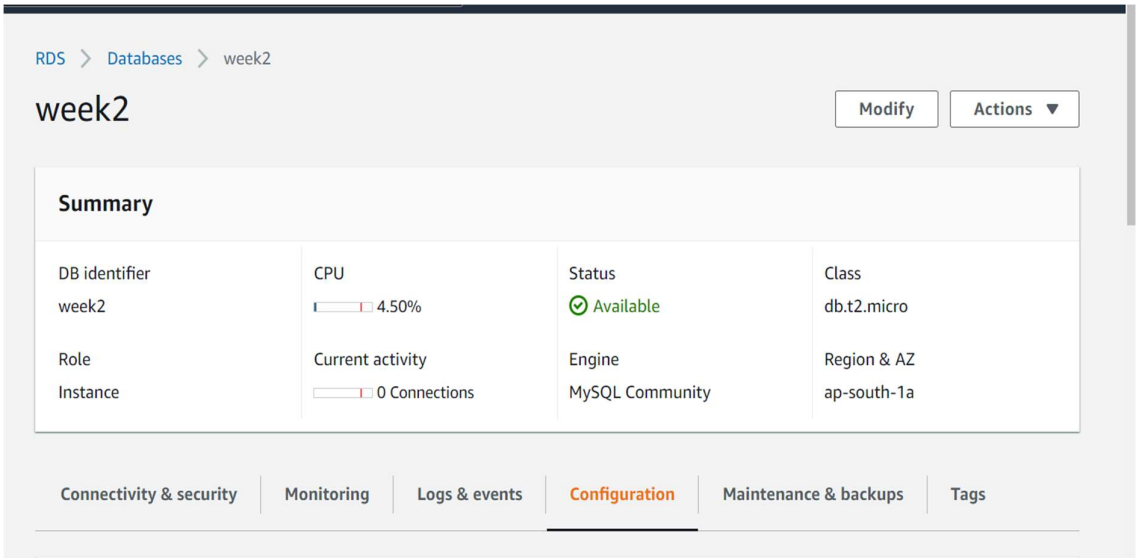
www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

Verifying the size after testing the lambda function . Here we can see that the image size has been compressed.

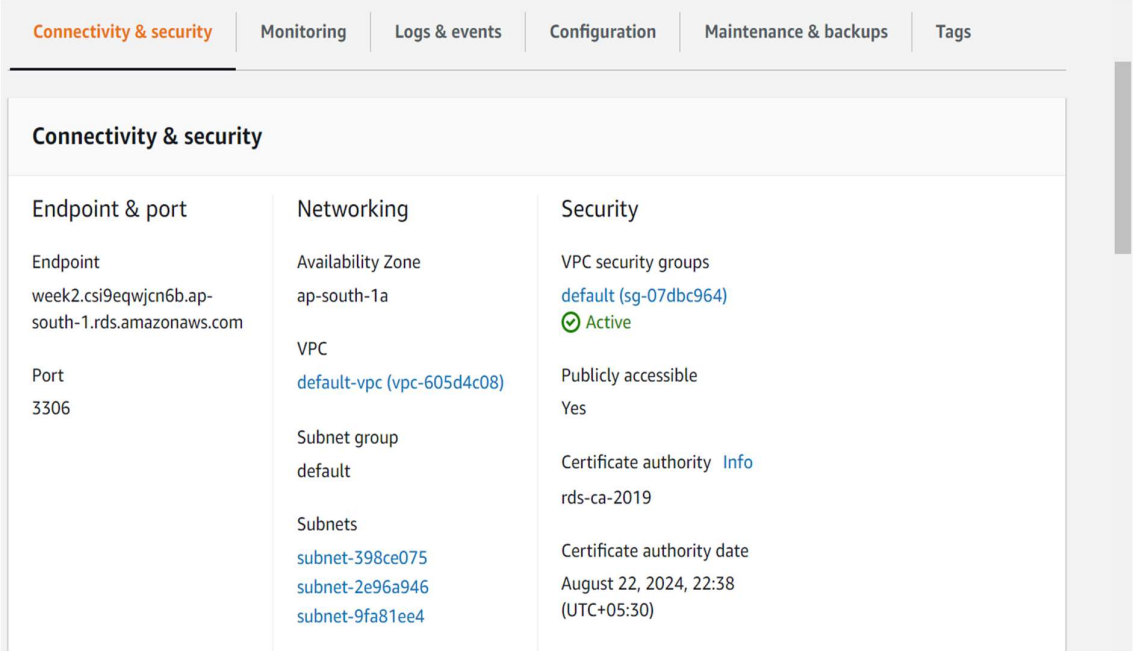


Task 2 Connect to a database in RDS using Python and query data using mysql-connector and Pandas. Save the queried data as a CSV file in S3. The SQL query might include Filter, GroupBy, and Aggregation clauses. [Each one of the interns will be getting a unique SQL scenario]

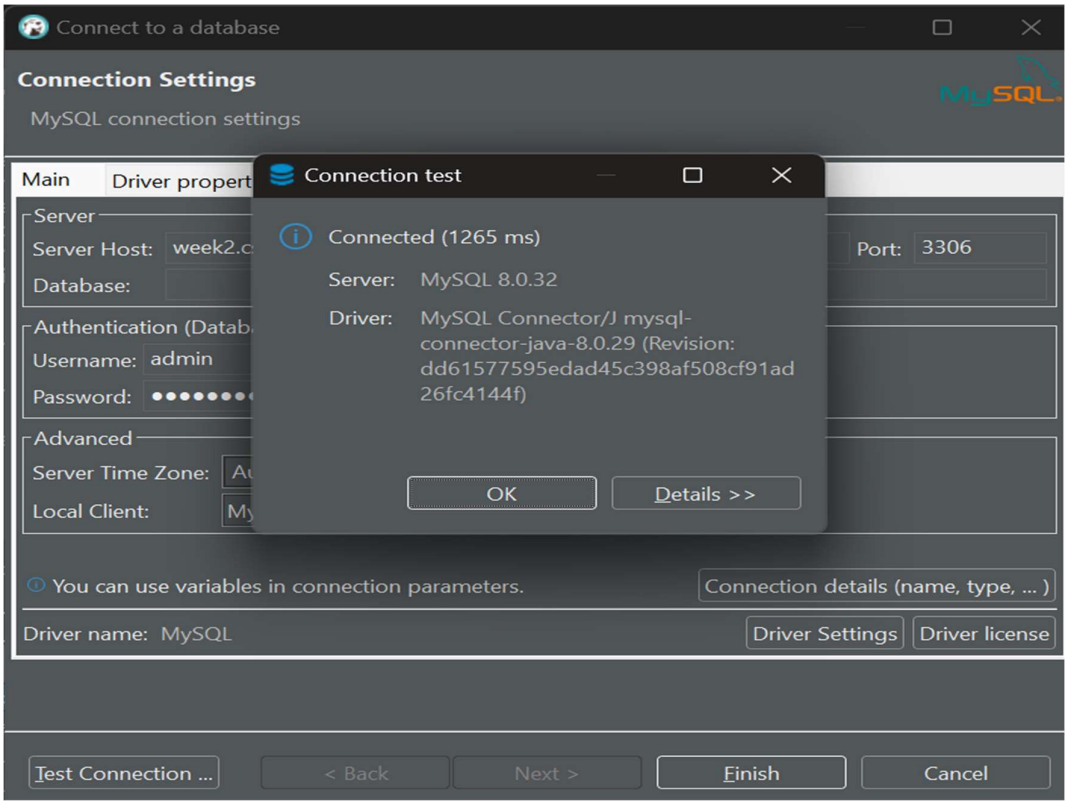
Step 1: Created an RDS MySQL server.



www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

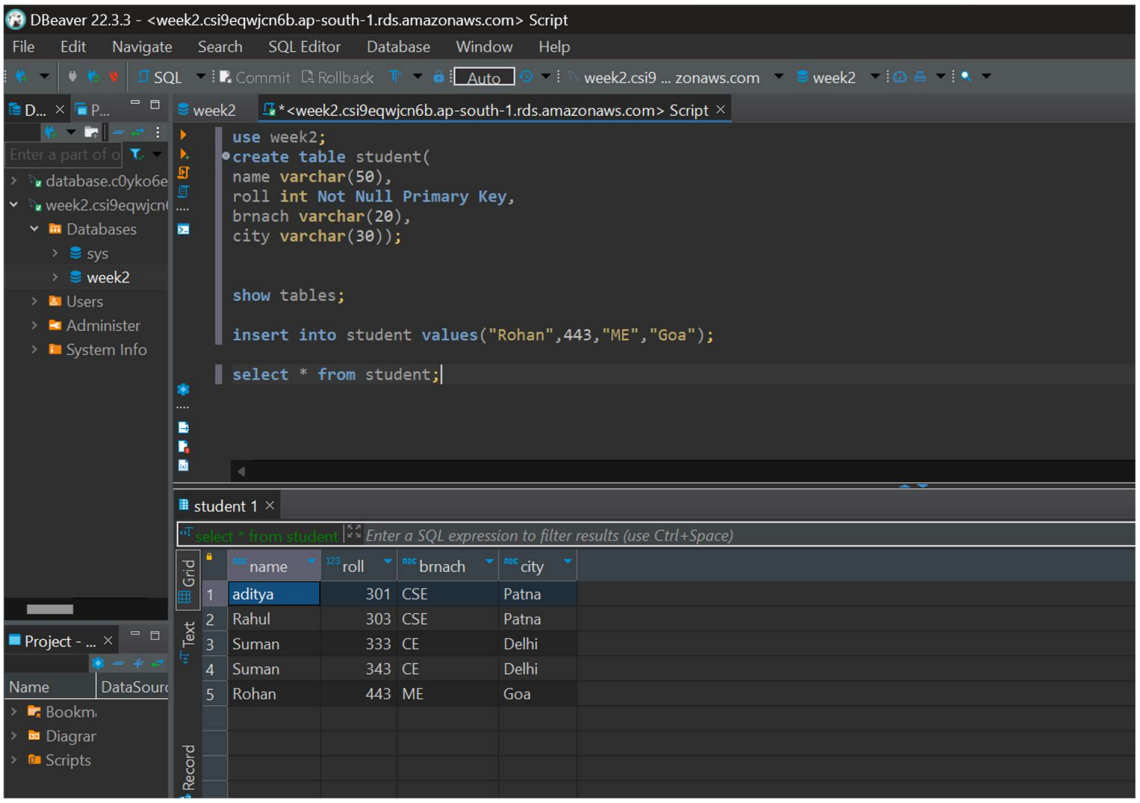


Step 2: Login that MySQL server using DBEVER software and also created a student table and adeed some data.

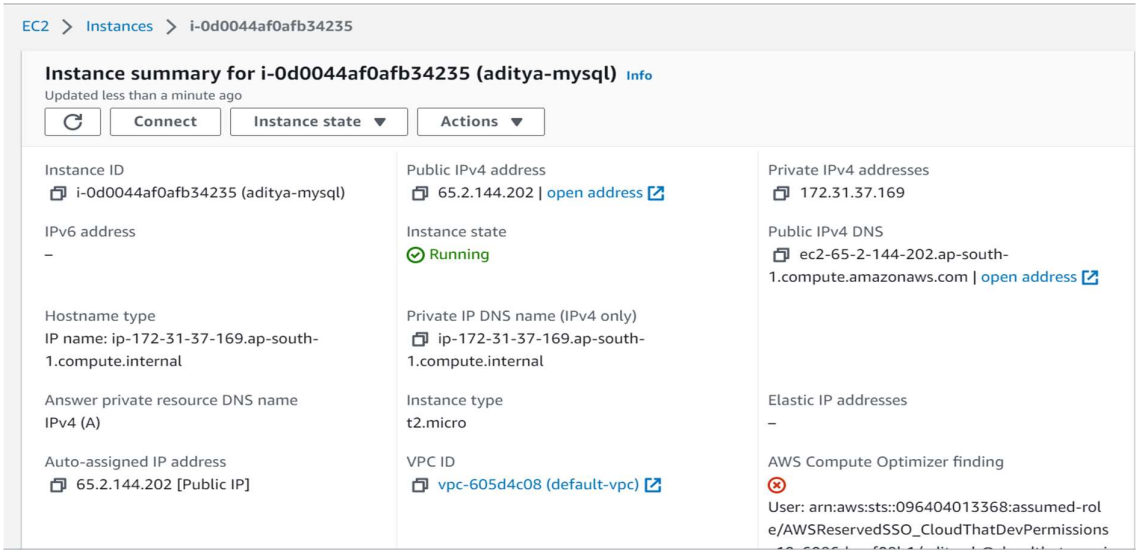


www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

Selecting the database and creating a table to add some into database of MySQL server. Also inserting some values on that table. After that viewing that data.



Step 3: Created a EC2 instance to perform query using pandas.



www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

Step 4: Login that instance using the SSH connection.

```
PS C:\Users\AdityaKumar\Downloads> ssh -i "adityanew.pem" ec2-user@ec2-65-2-144-202.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-2-144-202.ap-south-1.compute.amazonaws.com (65.2.144.202)' can't be established.
ED25519 key fingerprint is SHA256:UUiZQAQeJ3pmbiL/3+BY6Lja9Kw0TagtH4rhRlX1X+o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-2-144-202.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
```

Step 5: Update the ec2 instance and install python3 and pip and required python libraries.

Installing Updates

```
[ec2-user@ip-172-31-37-169 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package kernel.x86_64 0:5.10.177-158.645.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=================================================================================================================
 Package               Arch                Version                                Repository                  Size
=================================================================================================================
Installing:
 kernel                x86_64              5.10.177-158.645.amzn2                 amzn2extra-kernel-5.10      33 M

Transaction Summary
=================================================================================================================
Install  1 Package

Total download size: 33 M
Installed size: 136 M
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
kernel-5.10.177-158.645.amzn2.x86_64.rpm                                                          |  33 MB  00:00:00
```

Installing Python and checking their version.

```
[ec2-user@ip-172-31-37-169 ~]$ sudo yum install python3
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package python3-3.7.16-1.amzn2.0.2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-37-169 ~]$ python --version
Python 2.7.18
[ec2-user@ip-172-31-37-169 ~]$ python3 --versio
unknown option --versio
usage: python3 [option] ... [-c cmd | -m mod | file | -] [arg] ...
Try `python -h' for more information.
[ec2-user@ip-172-31-37-169 ~]$ python3 --version
Python 3.7.16
```

Installing pip and verifying it's version.

```
[ec2-user@ip-172-31-37-169 ~]$ python3 get-pip.py --user
Collecting pip
  Downloading pip-23.1-py3-none-any.whl (2.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 54.1 MB/s eta 0:00:00
Collecting wheel
  Downloading wheel-0.40.0-py3-none-any.whl (64 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.5/64.5 kB 13.8 MB/s eta 0:00:00
Installing collected packages: wheel, pip
Successfully installed pip-23.1 wheel-0.40.0
[ec2-user@ip-172-31-37-169 ~]$ pip --version
pip 23.1 from /home/ec2-user/.local/lib/python3.7/site-packages/pip (python 3.7)
[ec2-user@ip-172-31-37-169 ~]$ pip install mysql-connector
```

Installing MySQL-connecter and Pandas library to perform SQL queries using python.

www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

```
[ec2-user@ip-172-31-37-169 ~]$ pip install mysql-connector
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector
  Downloading mysql-connector-2.2.9.tar.gz (11.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.9/11.9 MB 63.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: mysql-connector
  Building wheel for mysql-connector (setup.py) ... done
  Created wheel for mysql-connector: filename=mysql_connector-2.2.9-cp37-cp37m-linux_x86_64.whl size=247950 sha256=0e10ea0be6e4be1f63
9f258ec39b60faf4fb05826a41adef9b4e6419f2729303
  Stored in directory: /home/ec2-user/.cache/pip/wheels/42/2f/c3/692fc7fc1f0d8c06b9175d94f0fc30f4f92348f5df5af1b8b7
Successfully built mysql-connector
Installing collected packages: mysql-connector
Successfully installed mysql-connector-2.2.9
[ec2-user@ip-172-31-37-169 ~]$ pip install pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Downloading pandas-1.3.5-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.3/11.3 MB 65.4 MB/s eta 0:00:00
Collecting python-dateutil>=2.7.3 (from pandas)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.7/247.7 kB 38.3 MB/s eta 0:00:00
Collecting pytz>=2017.3 (from pandas)
  Downloading pytz-2023.3-py2.py3-none-any.whl (502 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 502.3/502.3 kB 50.2 MB/s eta 0:00:00
Collecting numpy>=1.17.3 (from pandas)
  Downloading numpy-1.21.6-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.7/15.7 MB 56.2 MB/s eta 0:00:00
Collecting six>=1.5 (from python-dateutil>=2.7.3->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
```

Python script

```python
import mysql.connector
import pandas as pd

# establish database connection
db_connection = mysql.connector.connect(
  host="week2.csi9eqwjcn6b.ap-south-1.rds.amazonaws.com",
  user="admin",
  password="cloudthat",
  database="week2"
)

# define SQL query
sql_query = "select city ,count(*) as resident from student Group By city"

# execute SQL query and load results into a pandas dataframe
df = pd.read_sql(sql_query, con=db_connection)

# close database connection
db_connection.close()

# save dataframe as CSV file in memory
#csv_buffer = StringIO()
df.to_csv('query.csv', index=False)
print(df.head())
```

Step 6: Running the python script.

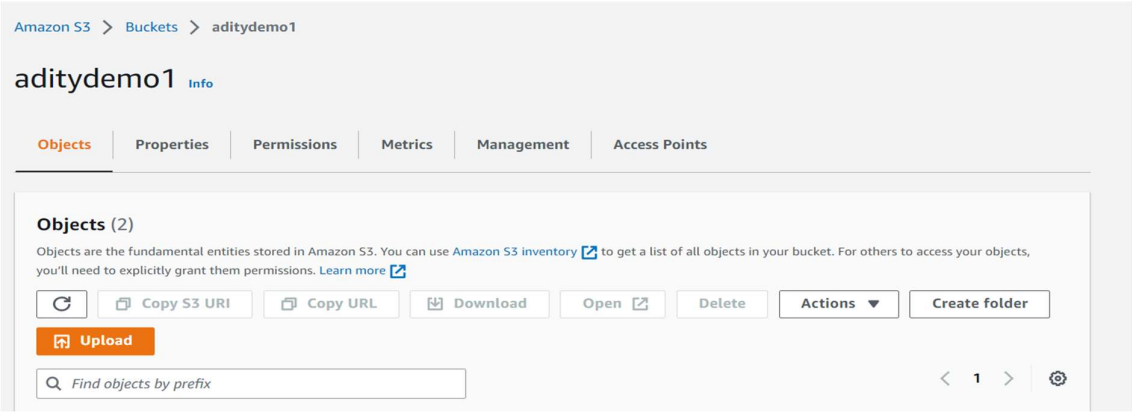
```
[ec2-user@ip-172-31-37-169 ~]$ python3 aditya.py
    city  resident
0  Patna         2
1  Delhi         2
2    Goa         1
[ec2-user@ip-172-31-37-169 ~]$
```


www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

after the successful execution of the python script, it saves the query into a csv file. Here we can see that it created by python script.

```
[ec2-user@ip-172-31-37-169 ~]$ ls
aditya.py  get-pip.py  __pycache__  query.csv
[ec2-user@ip-172-31-37-169 ~]$
```

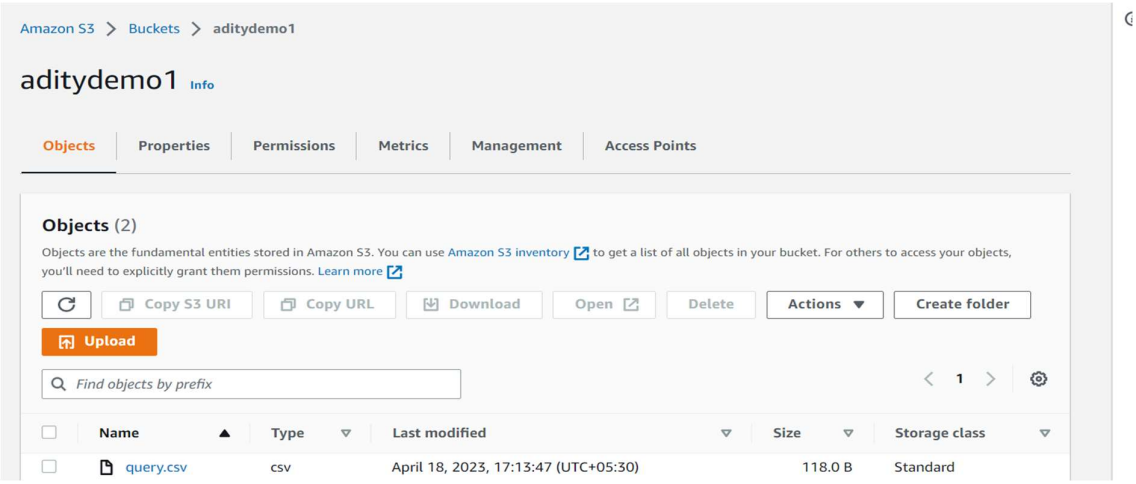
Step 7: Now we have to store the “Query.csv” file on S3 bucket. For that are going to create a S3 bucket.



Now we are storing the “Query.csv” file on S3 bucket using the AWS s3 copy command.

```
[ec2-user@ip-172-31-37-169 ~]$ aws s3 cp query.csv s3://aditydemo1
upload: ./query.csv to s3://aditydemo1/query.csv
```

Verifying the resources in s3 bucket.



www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.

Task 3: Create an IAM policy to get access to AWS resources.

Policies > recource_policy

Summary

Delete policy

Policy ARN arn:aws:iam::907191564732:policy/recource_policy

Description

Permissions | Policy usage | Tags | Policy versions | Access Advisor

Policy summary | { } JSON | Edit policy

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "*",
7              "Resource": "*"
8          }
9      ]
10 }
```


www.cloudthat.com
+91-888-000-2200
#610, 2nd Floor, 80 Feet Road, 6th Block
Koramangala Bengaluru – 560095.