

## Amazon Bucket

The screenshot shows the Amazon S3 console for the bucket 'mingruoqu1'. The left sidebar contains navigation links: Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, AWS Organizations settings), Feature spotlight, and AWS Marketplace for S3. The main content area shows the bucket's 'Objects' tab with a list of four objects:

Name	Type	Size	Storage class	ETag
exp1.csv	csv	20.0 B	Standard	f212b052857a86907b245f75ede72fca
exp2.csv	csv	60.0 B	Standard	c98f33bd5e5e2a46d367b2f39ce3ee63
exp3.csv	csv	105.0 B	Standard	1e1d46edd1ef54e7bc8ca46ced8fbd55
test	-	20.0 B	Standard	f212b052857a86907b245f75ede72fca

## Set up

The screenshot shows a Jupyter Notebook titled 'NoSQL-NoAWS' with the following Python code:

```
22.0.>=1.21.65->boto3) (1.26.4)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in ./opt/anaconda3/lib/python3.8/site-packages (from botocore<1.22.0,>=1.21.65->boto3) (2.8.1)
Requirement already satisfied: six>=1.5 in ./opt/anaconda3/lib/python3.8/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.65->boto3) (1.15.0)

In [55]: import boto3

In [56]: #In the first section, Python will check if you need to install Pycycopg2 library and install it.
import subprocess
import sys

def install(package):
    #if pip doesn't work, try pip3 in the following statement
    subprocess.check_call([sys.executable, "-m", "pip", "install", package])

install("boto3")

import boto3
import csv

In [57]: s3 = boto3.resource('s3',
    aws_access_key_id='xxx',
    aws_secret_access_key='xxx')

In [58]: try:
    s3.create_bucket(Bucket='mingruoqu1', CreateBucketConfiguration={
        'LocationConstraint': 'us-west-2'})
    except Exception as e:
        print (e)

In [60]: # make this bucket publicly readable. We will also need to make each blob in the bucket publicly readable
bucket = s3.Bucket('mingruoqu1')
bucket.Acl().put(ACL='public-read')
```

the query you use on your local machine to pull the data from your Dynamo DB

jupyter NoSQL 最新检查点: 2021年10月20日 (已自动保存)

```
'ReadCapacityUnits': 5,
'WriteCapacityUnits': 5
})
except Exception as e:
    print(e)
    #if there is an exception, the table may already exist. if so...
    table = dyndb.Table("DataTable1")

In [68]: table.meta.client.get_waiter('table_exists').wait(TableName='DataTable1')

In [69]: print(table.item_count)

0

In [70]: with open('/Users/qmr/Desktop/cmu_semester1/cloud/experiments.csv', 'r') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',', quotechar='|')
    next(csvf)
    for item in csvf:
        print(item)
        body = open('/Users/qmr/Desktop/cmu_semester1/cloud/'+ item[4], 'rb')
        s3.Object('mingruoqul', item[4]).put(Body=body)
        md = s3.Object('mingruoqul', item[4]).Acl().put(ACL='public-read')
        url = "https://s3.console.aws.amazon.com/s3/buckets/mingruoqul?region=us-west-2/"+item[4]
        metadata_item = {'PartitionKey': item[0], 'RowKey': item[1],
            'Conductivity': item[2], 'Concentration': item[3], 'url':url}
        try:
            table.put_item(Item=metadata_item)
        except:
            print("item may already be there or another failure")

['1', '-1', '52', '3.4', 'expl.csv']
['2', '-2', '52.1', '3.4', 'exp2.csv']
['3', '-2.93', '57.1', '3.7', 'exp3.csv']
```

Screenshot of the results of the above query from your local machine's terminal

jupyter NoSQL 最新检查点: 2021年10月20日 (更改未保存)

```
In [70]: with open('/Users/qmr/Desktop/cmu_semester1/cloud/experiments.csv', 'r') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',', quotechar='|')
    next(csvf)
    for item in csvf:
        print(item)
        body = open('/Users/qmr/Desktop/cmu_semester1/cloud/'+ item[4], 'rb')
        s3.Object('mingruoqul', item[4]).put(Body=body)
        md = s3.Object('mingruoqul', item[4]).Acl().put(ACL='public-read')
        url = "https://s3.console.aws.amazon.com/s3/buckets/mingruoqul?region=us-west-2/"+item[4]
        metadata_item = {'PartitionKey': item[0], 'RowKey': item[1],
            'Conductivity': item[2], 'Concentration': item[3], 'url':url}
        try:
            table.put_item(Item=metadata_item)
        except:
            print("item may already be there or another failure")

['1', '-1', '52', '3.4', 'expl.csv']
['2', '-2', '52.1', '3.4', 'exp2.csv']
['3', '-2.93', '57.1', '3.7', 'exp3.csv']

In [92]: response = table.get_item(
    Key={
        'PartitionKey': '1',
        'RowKey': '-1'
    }
)
item = response['Item']

In [93]: print(item)

{'Concentration': '3.4', 'PartitionKey': '1', 'RowKey': '-1', 'url': 'https://s3.console.aws.amazon.com/s3/buckets/mingruoqul?region=us-west-2/expl.csv', 'Conductivity': '52'}

In [94]: response2 = table.get_item(
```

```
In [93]: print(item)

{'Concentration': '3.4', 'PartitionKey': '1', 'RowKey': '-1', 'url': 'https://s3.console.aws.amazon.com/s3/buckets/mi
ngruoqul?region=us-west-2/exp1.csv', 'Conductivity': '52'}
```

```
In [94]: response2 = table.get_item(
        Key={
            'PartitionKey': '2',
            'RowKey': '-2'
        }
    )
item2 = response2['Item']
```

```
In [95]: print(item2)

{'Concentration': '3.4', 'PartitionKey': '2', 'RowKey': '-2', 'url': 'https://s3.console.aws.amazon.com/s3/buckets/mi
ngruoqul?region=us-west-2/exp2.csv', 'Conductivity': '52.1'}
```

```
In [96]: response3 = table.get_item(
        Key={
            'PartitionKey': '3',
            'RowKey': '-2.93'
        }
    )
item3 = response3['Item']
```

```
In [97]: print(item3)

{'Concentration': '3.7', 'PartitionKey': '3', 'RowKey': '-2.93', 'url': 'https://s3.console.aws.amazon.com/s3/bucket
s/mingruoqul?region=us-west-2/exp3.csv', 'Conductivity': '57.1'}
```

```
In [ ]:
```