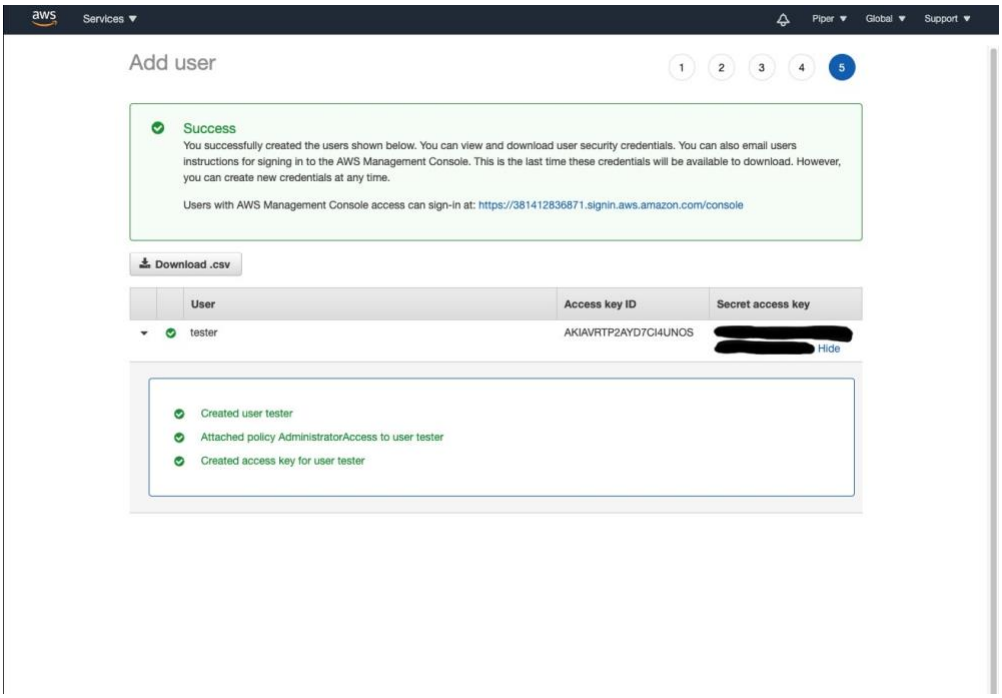
CS1660 Cloud Computing
HW#2
Piper Su

Part 1 – AWS account with IAM



Part 2 – Installed Amazon Python boto3 SDK

```
In [ ]: # !pip3 install boto3

In [72]: !pip3 list
Package         Version
--------------- -------
boto3           1.15.11
botocore        1.18.11
jmespath        0.10.0
pip             20.1.1
python-dateutil 2.8.1
s3transfer      0.3.3
setuptools      49.2.0
six             1.15.0
urllib3         1.25.10
wheel           0.34.2
```

Part 3 – code

In [3]:
```
import boto3
```

In [4]:
```
s3 = boto3.resource('s3',
 aws_access_key_id='AKIAVRTP2AYD7CI4UNOS',
 aws_secret_access_key='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'
)
```

In [5]:
```
try:
 s3.create_bucket(Bucket='datacont-piper', CreateBucketConfiguration={'LocationConstraint': 'us-west-2'})
except:
 print("this may already exist")
```

```
this may already exist
```

In [6]:
```
bucket = s3.Bucket("datacont-piper")
```

In [7]:
```
bucket.Acl().put(ACL='public-read')
```

Out[7]:
```
{'ResponseMetadata': {'RequestId': '883BBB4698B711E7',
  'HostId': 'S+3m3LmHDeOwr5iMu5+Ab1nM/aj8hGUs3OOrYvGp2V+3hjzkOc4qRT9RHQwDOFG8ucaYymtm0NI=',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amz-id-2': 'S+3m3LmHDeOwr5iMu5+Ab1nM/aj8hGUs3OOrYvGp2V+3hjzkOc4qRT9RHQwDOFG8ucaYymtm0NI=',
   'x-amz-request-id': '883BBB4698B711E7',
   'date': 'Mon, 05 Oct 2020 01:25:31 GMT',
   'content-length': '0',
   'server': 'AmazonS3'},
  'RetryAttempts': 0}}
```

In [8]:
```
#upload a new object into the bucket
body = open('/Users/sue/Public/cs1660_CC/test-s3-upload.java', 'rb')
```

In [9]:
```
o = s3.Object('datacont-piper', 'test').put(Body=body )
```

In [10]:
```
body = open('/Users/sue/Public/cs1660_CC/experiments.csv', 'rb')
o = s3.Object('datacont-piper', 'testcsv').put(Body=body )
```

In [11]:
```
s3.Object('datacont-piper', 'test').Acl().put(ACL='public-read')
s3.Object('datacont-piper', 'testcsv').Acl().put(ACL='public-read')
```

Out[11]:
```
{'ResponseMetadata': {'RequestId': '7E04D3D339292FD6',
  'HostId': 'D9dv/JxT2LWdT9yPJKHT78e9yuOae95bUE3wOYLg4RJC9pZv+KUe1SXKIeumHVekmhAu2fFOqDk=',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amz-id-2': 'D9dv/JxT2LWdT9yPJKHT78e9yuOae95bUE3wOYLg4RJC9pZv+KUe1SXKIeumHVekmhAu2fFOqDk=',
   'x-amz-request-id': '7E04D3D339292FD6',
   'date': 'Mon, 05 Oct 2020 01:25:31 GMT',
   'content-length': '0',
   'server': 'AmazonS3'},
  'RetryAttempts': 0}}
```

In [12]:
```
dyndb = boto3.resource('dynamodb',
 region_name='us-west-2',
 aws_access_key_id='AKIAVRTP2AYD7CI4UNOS',
 aws_secret_access_key='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'
)
```

In [13]:
```
try:
    table = dyndb.create_table(
        TableName='myTable1',
        KeySchema=[
            {
                'AttributeName': 'PartitionKey',
                'KeyType': 'HASH'
            },
            {
                'AttributeName': 'RowKey',
                'KeyType': 'RANGE'
            }
        ],
        AttributeDefinitions=[
            {
                'AttributeName': 'PartitionKey',
                'AttributeType': 'S'
            },
            {
                'AttributeName': 'RowKey',
                'AttributeType': 'S'
            },
        ],
        ProvisionedThroughput={
            'ReadCapacityUnits': 5,
            'WriteCapacityUnits': 5
        }
    )
except:
    #if there is an exception, the table may already exist. if so...
    table = dyndb.Table("myTable1")

table.meta.client.get_waiter('table_exists').wait(TableName='myTable1')
```

In [14]:
```
print(table.item_count)
```

```
0
```

In [15]:
```
import csv
```

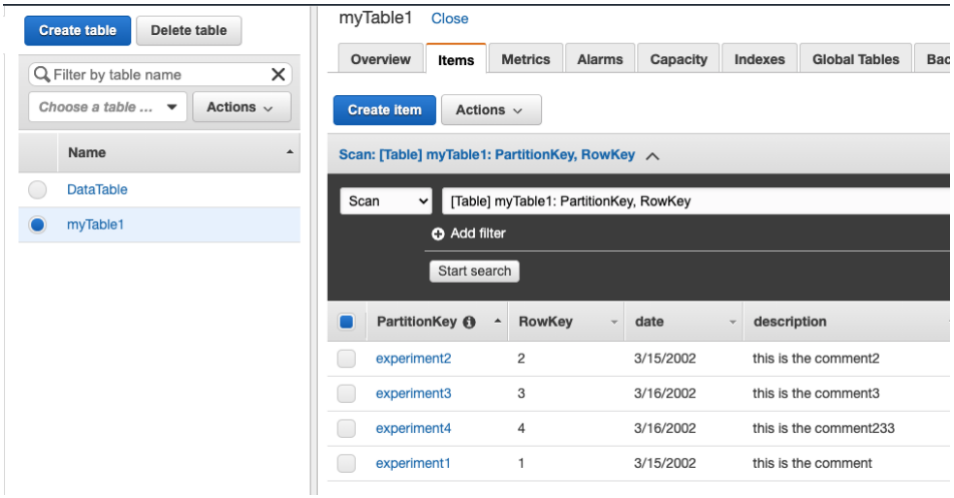
In [16]:
```
with open('/Users/sue/Public/cs1660_CC/experiments.csv', 'r', newline='') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',', quotechar='|')
    for item in csvf:
        print(item)

        metadata_item = {
            'PartitionKey': item[0],
            'RowKey': item[1],
            'description' : item[4],
            'date' : item[2]
        }

        try:
            table.put_item(Item=metadata_item)
        except:
            print("item may already be there or another failure")
```

```
['\ufeffexperiment1', '1', '3/15/2002', 'exp1', 'this is the comment']
['experiment2', '2', '3/15/2002', 'exp2', 'this is the comment2']
['experiment3', '3', '3/16/2002', 'exp3', 'this is the comment3']
['experiment4', '4', '3/16/2002', 'exp4', 'this is the comment233']
```

Part 4 – query to the newly constructed table



In [19]:
```
response = table.get_item(
 Key={
 'PartitionKey': 'experiment1',
 'RowKey': '1'
 }
)
```

In [20]:
```
response
```

Out[20]:
```
{'ResponseMetadata': {'RequestId': 'I1N0IT4RO06KGGM0VNMT9OGONFVV4KQNSO5AEMVJF66Q9ASUAAJG',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'server': 'Server',
   'date': 'Mon, 05 Oct 2020 01:32:02 GMT',
   'content-type': 'application/x-amz-json-1.0',
   'content-length': '2',
   'connection': 'keep-alive',
   'x-amzn-requestid': 'I1N0IT4RO06KGGM0VNMT9OGONFVV4KQNSO5AEMVJF66Q9ASUAAJG',
   'x-amz-crc32': '2745614147'},
  'RetryAttempts': 0}}
```