Lab. AWS DynamoDB CRUD with Python boto3

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/dynamodb.html
https://www.dynamodbguide.com/
https://docs.aws.amazon.com/
https://onikaze.tistory.com/603
https://skyseven73.tistory.com/18

1. IAM User Create for AWS Service

1)사용자 이름 : dynamodb-developer
2)권한 : AmazonDynamoDBFullAccess
3)보안 자격 증명에서 액세스 키 생성
  -사용사례 : AWS 컴퓨팅 서비스에서 실행되는 애플리케이션
  -"위의 권장 사항을 이해했으며 액세스 키 생성을 계속하려고 합니다." Check
4)Access Key의 CSV 파일 저장

2. AWS CLI Configure

```
(PythonRoom) D:\PythonRoom>aws configure
AWS Access Key ID [****************2KDZ]: AKIA3PU7RNERREPAAIOT
AWS Secret Access Key [****************fhCQ]: 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
Default region name [ap-northeast-2]: ap-northeast-2
Default output format [json]: json
```

3. DynamoDB Table Create

```
> aws dynamodb create-table --table-name Movie --attribute-definitions        AttributeName=Code,AttributeType=S
AttributeName=Name,AttributeType=S --key-schema          AttributeName=Code,KeyType=HASH
AttributeName=Name,KeyType=RANGE --provisioned-throughput          ReadCapacityUnits=5,WriteCapacityUnits=5
--table-class STANDARD
```

4. boto3 Install in Python Virtual Environment

```
> pip list
> python -m pip install --upgrade pip
> pip install boto3
```

5. Resource Object Create

```
import boto3

resource = boto3.resource('dynamodb', region_name='ap-northeast-2',
                      aws_access_key_id = 'AKIA3PU7RNERREPAAIOT',
                      aws_secret_access_key = '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')

client = boto3.client('dynamodb')
```

6. Table List

```
client.list_tables()

{'TableNames': ['Movie'],
 'ResponseMetadata': {'RequestId': '795OLMT756UQMM9LR42SA673E7VV4KQNSO5AEMVJF66Q9ASUAAJG',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'server': 'Server',
   'date': 'Thu, 20 Jul 2023 23:59:06 GMT',
   'content-type': 'application/x-amz-json-1.0',
   'content-length': '24',
   'connection': 'keep-alive',
   'x-amzn-requestid': '795OLMT756UQMM9LR42SA673E7VV4KQNSO5AEMVJF66Q9ASUAAJG',
   'x-amz-crc32': '3057459838'},
  'RetryAttempts': 0}}
```

7. Movie Table Information

```
table = client.describe_table(
    TableName='Movie'
)
print(table)

{'Table': {'AttributeDefinitions': [{'AttributeName': 'Code', 'AttributeType': 'S'}, {'AttributeName': 'Name', 'AttributeType':
'S'}], 'TableName': 'Movie', 'KeySchema': [{'AttributeName': 'Code', 'KeyType': 'HASH'}, {'AttributeName': 'Name', 'KeyType':
'RANGE'}], 'TableStatus': 'ACTIVE', 'CreationDateTime': datetime.datetime(2023, 7, 21, 8, 9, 17, 255000, tzinfo=tzlocal()),
'ProvisionedThroughput': {'NumberOfDecreasesToday': 0, 'ReadCapacityUnits': 5, 'WriteCapacityUnits': 5}, 'TableSizeBytes': 0,
'ItemCount': 0, 'TableArn': 'arn:aws:dynamodb:ap-northeast-2:789534828835:table/Movie', 'TableId':
'3d0fdcf0-1137-4bb3-a088-e755113b1da1', 'TableClassSummary': {'TableClass': 'STANDARD'}, 'DeletionProtectionEnabled':
False}, 'ResponseMetadata': {'RequestId': 'RIAFSTMTI61PLEDS63QI4OTK3BVV4KQNSO5AEMVJF66Q9ASUAAJG',
'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Fri, 21 Jul 2023 00:00:58 GMT', 'content-type':
'application/x-amz-json-1.0', 'content-length': '634', 'connection': 'keep-alive', 'x-amzn-requestid':
'RIAFSTMTI61PLEDS63QI4OTK3BVV4KQNSO5AEMVJF66Q9ASUAAJG', 'x-amz-crc32': '1620184737'}, 'RetryAttempts': 0}}
```

8. Item Insert

```
table = resource.Table('Movie')
```

```
item = {'Code':'19780080', 'Name':'Star Wars', 'Genre':'SF',
    'Date':'1978-04-12', 'Director':'George Lucas',
    'Actor':'마크 해밀, 캐리 피셔, 해리슨 포드, 알렉 기네'}
table.put_item(Item=item)
item = {'Code':'20050112', 'Name':'Batman Begins', 'Running Time' : 134, 'Genre':'범죄, 액션, 판타지', 'Date':'2005-06-24', 'Director':'크리스토퍼 놀란', 'Actor':'리암 니슨, 크리스찬 베일, 마이클 케인'}
table.put_item(Item=item)
```

9. Item 모두 가져오기

```
results = table.scan()
items = results['Items']
count = results['Count']
print(items)
print(count)
```

10. Item 검색하기

1)get_item() 이용하기

```
    response = table.get_item(
        Key={
            'Code': '20050112',
            'Name': 'Batman Begins'
        }
    )
    item = response['Item']
    print(item)
```

2)query() 이용하기

```
    from boto3.dynamodb.conditions import Key

    query = {"KeyConditionExpression": Key("Code").eq("20050112")}
    print(table.query(**query))
```

3)scan() 이용하기

```
    from boto3.dynamodb.conditions import Attr

    query = {"FilterExpression": Attr('Name').eq('Batman Begins')}
    response = table.scan(**query)
    print(response['Items'])
```

11. Item Update

```
table.update_item(
    Key={
        'Code': '20050112',
        'Name': 'Batman Begins'
    },
    UpdateExpression='SET Director = :myvariable',
    ExpressionAttributeValues={
        ':myvariable': 'Christopher Nolan'
    }
)

response = table.get_item(
    Key={
        'Code': '20050112',
        'Name': 'Batman Begins'
    }
)
item = response['Item']
print(item)
```

12. Item Delete

```
table.delete_item(
    Key={
        'Code': '20050112',
        'Name': 'Batman Begins'
    }
)
```

13. Table Delete

```
table.delete()
```