

Joy Kaufman

November 13, 2020

SDEV 400 Secure Programming in the Cloud

Homework 2

1. Part I

1.1 Creating the sensor table, shown in Figure 1

```
(venv) vocstartsoft:~/environment $  
(venv) vocstartsoft:~/environment $ aws dynamodb create-table \  
> --table-name Sensors \  
> --attribute-definitions \  
>   AttributeName=Sensor,AttributeType=S \  
> --key-schema \  
>   AttributeName=Sensor,KeyType=HASH \  
> --provisioned-throughput \  
>   ReadCapacityUnits=5,WriteCapacityUnits=5  
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Sensor",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "Sensors",  
    "KeySchema": [  
      {  
        "AttributeName": "Sensor",  
        "KeyType": "HASH"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": 1605375429.853,  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "ReadCapacityUnits": 5,  
      "WriteCapacityUnits": 5  
    },  
    "TableSizeBytes": 0,  
    "ItemCount": 0,  
    "TableArn": "arn:aws:dynamodb:us-east-1:730601335170:table/Sensors",  
    "TableId": "6a47e812-f776-444e-9fc0-ce5dbd532d38"  
  }  
}
```

[Figure 1]

1.2 Uploading sensor records from a JSON file using the AWS CLI, first one then multiple, shown in figures 2 and 3 respectively. (I copy + cut all the additional records out of the Sensors.json file on the first run, then put them back for the second.)

```
(venv) vocstartsoft:~/environment $ aws dynamodb put-item --table-name Sensors --item file://week_two/sensors.js  
on --return-consumed-capacity TOTAL  
{  
  "ConsumedCapacity": {  
    "TableName": "Sensors",  
    "CapacityUnits": 1.0  
  }  
}  
(venv) vocstartsoft:~/environment $
```

[Figure 2]

```

    "Locations": { "L": [{ "S": "Narnia NZ" }, { "S": "Aberdeen MD" }, { "S": "Durham NC" } ] }

(env) vocstartsoft:~/environment $ aws dynamodb batch-write-item --request-items file://week_two/sensors.json
{
  "UnprocessedItems": {}
}
(env) vocstartsoft:~/environment $

```

[Figure 3]

1.3 Printed out a representation of all the sensors now in the table, shown in Figure 4.

```

bash - "ip-172-31-0-98" x week_one/s_three.py - Stc x week_one/s_three.py - Stc x [New] - Stopped
(env) vocstartsoft:~/environment/week_two $ aws
(env) vocstartsoft:~/environment/week_two $ aws dynamodb scan --table-name Sensors
{
  "Items": [
    {
      "SensorDescription": {
        "S": "I Sensor"
      },
      "Sensor": {
        "S": "I"
      },
      "ImageFile": {
        "S": "/Sensors/Images/I.png"
      }
    },
    {
      "SensorDescription": {
        "S": "A Sensor"
      },
      "Sensor": {
        "S": "A"
      },
      "ImageFile": {
        "S": "/Sensors/Images/A.png"
      }
    },
    {
      "SensorDescription": {

```

[Figure 4]

2. Part II

2.1 As shown in Figure 5, created the Courses table.

```
14     AttributeDefinitions = [
15         {
16             "AttributeName": "CourseID",
17             "AttributeType": "S"
18         }
19     ],
20     ProvisionedThroughput = {
21         'ReadCapacityUnits': 10,
22         'WriteCapacityUnits': 10
23     }
24 )
25
26 return course_tbl
27
28 if __name__ == "__main__":
29     course_tbl = build_course_tbl()
30     print(course_tbl)
```

18:14 Python Spaces: 4

bash - "p-172-31-5-99" × week_one/s_three.py - Stc × week_one/s_three.py - Stc × [New] - Stopped ×

(venv) vocstartsoft:~/environment/week_two \$ python app.py
dynamodb.Table(name='Courses')
(venv) vocstartsoft:~/environment/week_two \$

[Figure 5]

2.2 Uploaded records for our new table as shown in Figure 6. In retrospect it may have been faster to use the python syntax demoed in some of the course materials for this week, but I kept expecting one of the tasks to be a gotcha because of the warning that it would take longer than I expected. So since the other method of uploading records (CLI) had already worked for me I did it that way to minimize unknowns.

```
100     }
101   },
102   {
103     "PutRequest": {
104       "Item": {
105         "CourseID": { "S": "0010"},
106         "Subject": { "S": "WRTG"},
107         "CatalogNbr": { "N": "425"},
108         "Title": { "S": "Survey of Writing Styles"},
109         "NumCredits": { "N": "3" }
110       }
111     }
112   }
113 ]
114 }
115
```

115:1 JSON Spaces: 4

python3.6 - "p-172-31-5" x week_one/s_three.py - Stc x week_one/s_three.py - Stc x [New] - Stopped x +

```
(venv) vocstartsoft:~/environment/week_two $ aws dynamodb batch-write-item --request-items file:///courses.json
{
  "UnprocessedItems": {}
}
(venv) vocstartsoft:~/environment/week_two $
```

[Figure 6]

2.2 In Figure 7 a normal run through the program is demonstrated with the user entering valid search criteria, then exiting. Figure 8 details the program continuing to solicit valid input until it gets search criteria that it can use.

```
70 )
71
72 response = list(
73     filter(lambda x: x['CatalogNbr'] == catalog_no, response['Items'])
74 )
75
76 response_title = response[0]['Title']
77
78 print(f'The title of your selected course is: {response_title}')
79 except Exception as e:
80     print(f'Sorry, no matches found for these search criteria: {e} \n\n")
81
82
83 # Process logic if they want to perform additional searches
```

python - "p-172-31-5-98" x week_one/s_three.py - Stx week_one/s_three.py - Stx [New] - Stopped

```
(venv) vocstartsoft:~/environment/week_two $ python app.py
File "app.py", line 78
    print(f'The title of your selected course is: {response_title}')
                                     ^
SyntaxError: invalid syntax
(venv) vocstartsoft:~/environment/week_two $ python app.py
Welcome to the course catalog.

Please enter the 2-4 letter subject abbreviation: SDEV
Please enter the numeric course number: 480
The title of your selected course is: Secure Programming in the Cloud
Would you like to search for another title? (Y or N)Y
Please enter the 2-4 letter subject abbreviation: SDEV
Please enter the numeric course number: 350
The title of your selected course is: Building Secure Web Apps
Would you like to search for another title? (Y or N)N

Thank you for using the course catalog.
(venv) vocstartsoft:~/environment/week_two $
```

[Figure 7]

```
python - "p-172-31-5-98" x week_one/s_three.py - S x week_one/s_three.py - S x [New] - Stopped
```

```
(venv) vocstartsoft:~/environment/week_two $ python app.py
Welcome to the course catalog.

Please enter the 2-4 letter subject abbreviation:
Please enter the 2-4 letter subject abbreviation:
Please enter the 2-4 letter subject abbreviation: SDEV
Please enter the numeric course number:
Error gathering input: invalid literal for int() with base 10: ''
Please enter the numeric course number:
Error gathering input: invalid literal for int() with base 10: ''
Please enter the numeric course number: AB
Error gathering input: invalid literal for int() with base 10: 'AB'
Please enter the numeric course number: 300
Sorry, no matches found for these search criteria: list index out of range

Would you like to search for another title? (Y or N)
```

[Figure 8]