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

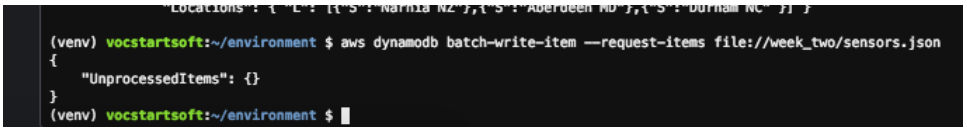
```
(venv) vocstartsoft:~/environment $
(venv) vocstartsoft:~/environment $ aws dynamodb create-table \
>    --table-name Sensors \
>    --attribute-definitions \
>       AttributeName=Sensor,AttributeType=S \
>    --key-schema \
>       AttributeName=Sensor,KeyType=HASH \
>    --provisioned-throughput \
>       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 our of the Sensors.json file on the first run, then put them back for the second.)

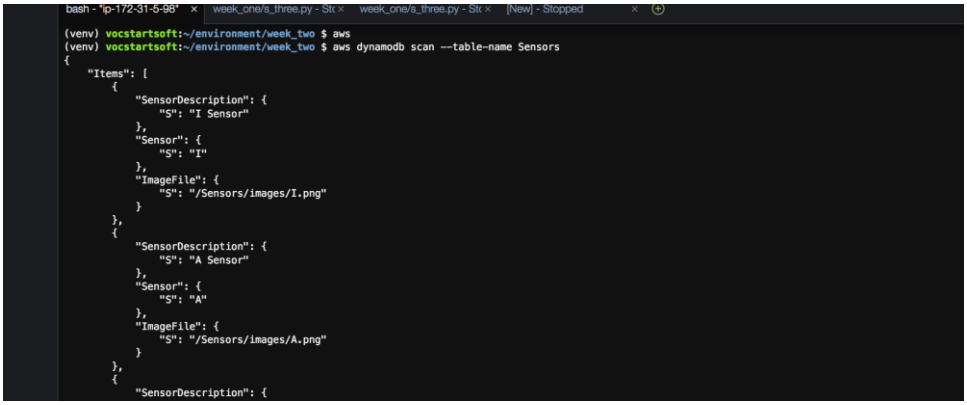
```
(venv) vocstartsoft:~/environment $ aws dynamodb put-item --table-name Sensors --item file://week_two/sensors.js
on --return-consumed-capacity TOTAL
{
    "ConsumedCapacity": {
        "TableName": "Sensors",
        "CapacityUnits": 1.0
    }
}
(venv) vocstartsoft:~/environment $
```

[Figure 2]



[Figure 3]

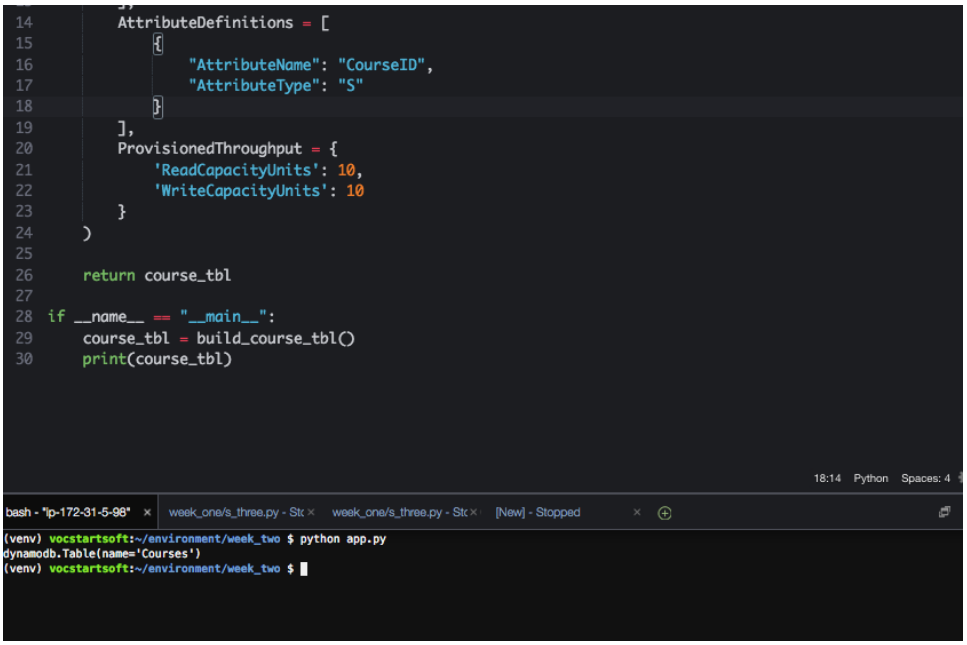
1.3 Printed out a representation of all the sensors now in the table, shown in Figure 4.



[Figure 4]

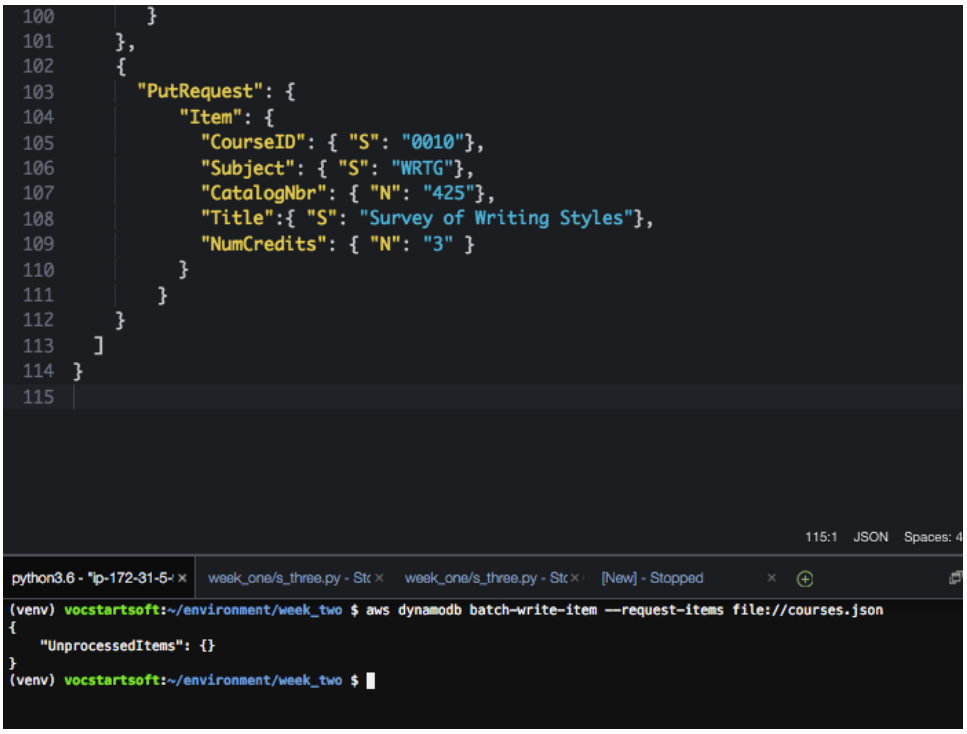
2. Part II

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



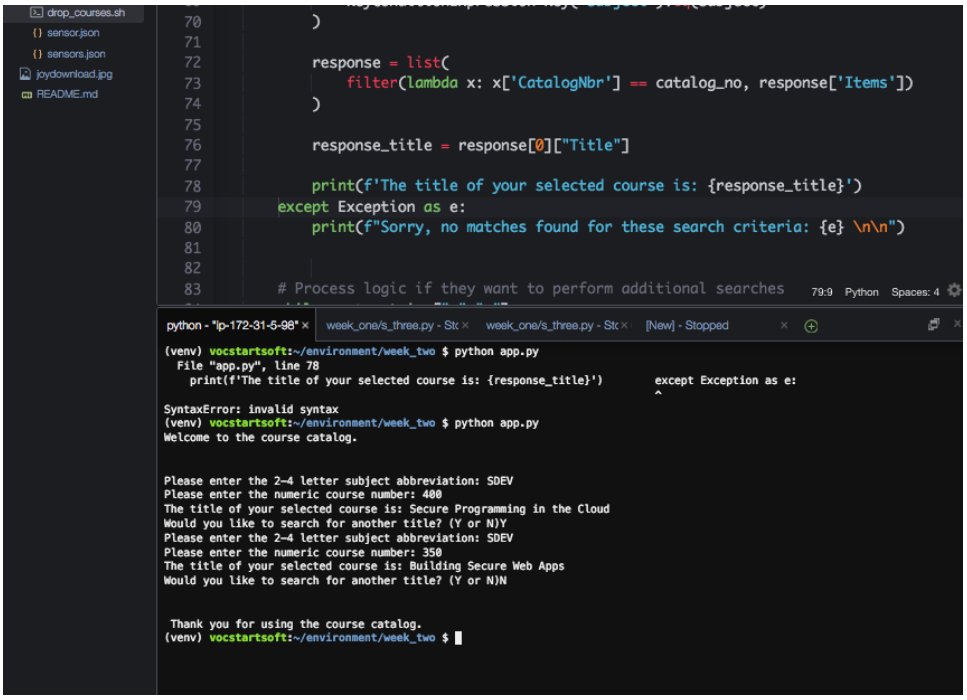
[Figure 5]

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.

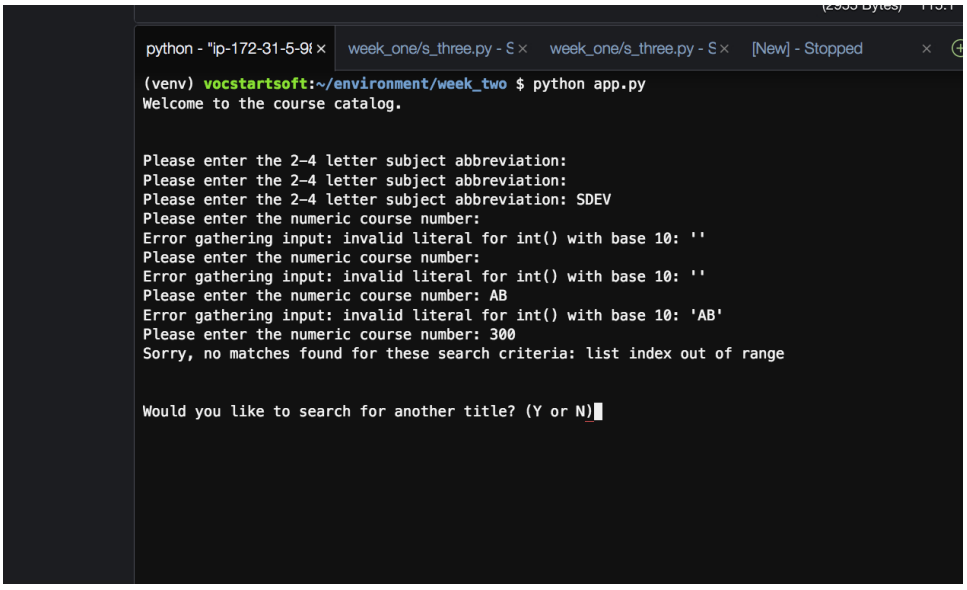


[Figure 6]

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.



[Figure 7]



[Figure 8]