**Assignment 5 - Introduction to Amazon Web Services (and Web Interface)**

**Name: Shweta Pathak**

**UTA ID: 1001154572**

**Net Id: ssp4572**

**Section: 1.00 – 3.00**

**References:**

1.[http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-common-steps.html](http://www.google.com/url?q=http%3A%2F%2Fdocs.aws.amazon.com%2Felasticbeanstalk%2Flatest%2Fdg%2Fcreate-deploy-python-common-steps.html&sa=D&sntz=1&usg=AFQjCNGBvsjGhvFdOm3PCiOrTz7bdMcKig)  
2.[http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-flask.html](http://www.google.com/url?q=http%3A%2F%2Fdocs.aws.amazon.com%2Felasticbeanstalk%2Flatest%2Fdg%2Fcreate-deploy-python-flask.html&sa=D&sntz=1&usg=AFQjCNGaUfy53pCSrMohUljWxgCo4Sjkfw)  
3.[http://boto.readthedocs.org/en/latest/dynamodb\_tut.html](http://www.google.com/url?q=http%3A%2F%2Fboto.readthedocs.org%2Fen%2Flatest%2Fdynamodb_tut.html&sa=D&sntz=1&usg=AFQjCNF04Sis7MzkMm67HGNOXzO86L9Rzg)  
4.[http://programminggenin.blogspot.com/2012/10/getting-started-with-amazons-dynamodb.html](http://www.google.com/url?q=http%3A%2F%2Fprogramminggenin.blogspot.com%2F2012%2F10%2Fgetting-started-with-amazons-dynamodb.html&sa=D&sntz=1&usg=AFQjCNHpDkUmTPgO3Y7Be4MwuRcx7OfwsA)  
5.[http://stackoverflow.com/questions/24665583/correct-boto-aws-dynamodb-scan-syntax](http://www.google.com/url?q=http%3A%2F%2Fstackoverflow.com%2Fquestions%2F24665583%2Fcorrect-boto-aws-dynamodb-scan-syntax&sa=D&sntz=1&usg=AFQjCNEhscr55ioIIsQppO8tUnOcXzqusw)

**Code:**

#Code to create database in Dynamo DB and insert data into it

# import statements

import boto

import csv

import time

import sys

import urllib2

import boto.dynamodb

from boto.s3.key import Key

from boto.s3.connection import S3Connection

from boto.s3.connection import Location

from boto.dynamodb2.fields import HashKey

from boto.dynamodb2.table import Table

from boto.dynamodb2.types import NUMBER

# Upload the file to amazon S3

def put():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.create\_bucket('shwetabucket91')

k = Key(bucket\_name)

k.key = raw\_input("Enter the file name to upload to S3: ")

start\_time = time.clock()

k.set\_contents\_from\_filename(k.key)

end\_time = time.clock()

total\_time = end\_time-start\_time

print(total\_time)

# Inserting data into the Relational database service

def createTable():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

connectDynamo = boto.dynamodb.connect\_to\_region('us-west-2',aws\_access\_key\_id=AWS\_ACCESS\_KEY\_ID,aws\_secret\_access\_key=AWS\_SECRET\_ACCESS\_KEY)

# Creating a table in Dynamo DB

#Schema creation

print "\nCreating a table ........"

inpatient\_table\_schema = connectDynamo.create\_schema(hash\_key\_name='Id',hash\_key\_proto\_value=str)

inpatient\_table = connectDynamo.create\_table(name = 'Inpatient\_data',schema = inpatient\_table\_schema,read\_units=10,write\_units=50)

#Listing the tables created

print "\nList of tables created :"

tables = connectDynamo.list\_tables()

print (tables[0:])

# Inserting data into the Dynamo DB table

def insert\_data():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.get\_bucket('shwetabucket91')

k = Key(bucket\_name)

# Reading the file from amazon S3 bucket

url = 'https://s3.amazonaws.com/shwetabucket91/data1.csv'

response = urllib2.urlopen(url)

csv\_data = csv.reader(response)

# Establish a connection with dynamodb2

connectDynamo = boto.dynamodb.connect\_to\_region('us-west-2',aws\_access\_key\_id=AWS\_ACCESS\_KEY\_ID,aws\_secret\_access\_key=AWS\_SECRET\_ACCESS\_KEY)

print "\nInserting to data into table .... "

table = connectDynamo.get\_table('Inpatient\_data')

count = 0

hashKey\_count = 1

start\_time = time.clock()

for row in csv\_data:

count += 1

hashKey\_count += 1

key = 'Id'+ str(hashKey\_count)

item\_data = {'DRG\_Definition': row[0],

'Provider\_Id': row[1],

'Provider\_Name': row[2],

'Address': row[3],

'City': row[4],

'State': row[5],

'Zip': row[6],

'Region': row[7],

'Total\_discharge': row[8],

'Average\_Covered\_Charges': row[9],

'Average\_Total\_Payments': row[10],

'Average\_Medicare\_Payments': row[11]}

item = table.new\_item(hash\_key = key ,attrs = item\_data)

item.put()

print count

end\_time = time.clock()

total\_time = end\_time - start\_time

print "\n Total time taken to insert data into Dynamo DB :"

print(total\_time)

# Listing all the buckets available

def list\_tables():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

connectDynamo = boto.dynamodb.connect\_to\_region('us-west-2',aws\_access\_key\_id=AWS\_ACCESS\_KEY\_ID,aws\_secret\_access\_key=AWS\_SECRET\_ACCESS\_KEY)

#Listing the tables created

print "\nList of tables created :"

tables = connectDynamo.list\_tables()

print (tables[0:])

# Query data stored in Dynamo DB

def main():

options\_toselect = {1: put, 2: createTable, 3:insert\_data,4:list\_tables}

while(True):

print "\n1. Upload file on Amazon Cloud S3. \n"

print "2. Create table into Amazon Dynamo DB \n"

print "3. Insert data into Amazon Dynamo DB table \n"

print "4. List tables created on Dynamo DB. \n"

print "5. Exit \n"

option = raw\_input("Select one option : ")

if option =="1":

options\_toselect[1]()

elif option =="2":

options\_toselect[2]()

elif option =="3":

options\_toselect[3]()

elif option =="4":

options\_toselect[4]()

elif option =="5":

sys.exit(0)

else:

print "Please select a valid choice !!!\n"

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Output:**

**Time taken to upload the file to Amazon S3: 0.018 seconds**

**Time taken to insert data into Dynamo DB: 85.78 seconds**

**# Code to create a web interface to take input from user and calculate time to return data**

**Application.py:**

from flask import Flask, render\_template, request, url\_for

from flask import Flask

from flask.ext.dynamo import Dynamo

import boto

import time

from boto.dynamodb.layer1 import Layer1

from boto.dynamodb import condition

# Initialize the Flask application

application = Flask(\_\_name\_\_)

# Define a route for the default URL, which loads the form

@application.route('/')

def form\_input():

return render\_template('form\_submit.html')

@application.route('/GetData/', methods=['POST'])

def GetData():

start\_time = time.clock()

Provider\_Id = request.form['id']

conn = boto.dynamodb.connect\_to\_region('us-west-2',aws\_access\_key\_id='AKIAJGQE6BZY4X7LYAYA',aws\_secret\_access\_key='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')

table = conn.get\_table('Inpatient\_data')

item = table.scan(scan\_filter={'Provider\_Id': boto.dynamodb.condition.EQ(Provider\_Id)})

items\_list=[]

for i in item:

items\_list.append(i)

end\_time=time.clock()

total = end\_time - start\_time

return render\_template('form\_action.html',time=total,list\_item=items\_list)

# Run the application

if \_\_name\_\_ == '\_\_main\_\_':

application.run(

host="0.0.0.0",

port=int("80")

)

**Form\_submit.html:**

<html>

<head>

<title>Assignment 5 - Introduction to Amazon Web Services</title>

<link rel=stylesheet type=text/css href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

<div id="container">

<div id="content">

<h1> Welcome to the Patient data system</h1><br />

<form method="post" action="{{ url\_for('GetData') }}">

<b> Provide a Provider\_Id :</b>

<input type="text" name="id" />

<input type="submit" name ="Get Results"/>

</form>

</div>

</div>

</body>

</html>

**Form\_action.html:**

<html>

<head>

<title>Patient details are :</title>

</head>

<body>

<div id="container">

<div id="content">

<b> Time taken to load :{{time}}</b><br /><br /><br />

</div>

<table border=2>

<tr>

<td>Id</td>

<td>DRG\_Definition</td>

<td>Provider\_Id</td>

<td>Provider\_Name</td>

<td>Address</td>

<td>City</td>

<td>State</td>

<td>Zip</td>

<td>Region</td>

<td>Total\_discharge</td>

<td>Average\_Covered\_Charges</td>

<td>Average\_Total\_Payments</td>

<td>Average\_Medicare\_Payments</td></tr>

<tr>{% for row in list\_item %}

<td>{{row['Id']}}</td>

<td>{{row['DRG\_Definition']}}</td>

<td>{{row['Provider\_Id']}}</td>

<td>{{row['Provider\_Name']}}</td>

<td>{{row['Address']}}</td>

<td>{{row['City']}}</td>

<td>{{row['State']}}</td>

<td>{{row['Zip']}}</td>

<td>{{row['Region']}}</td>

<td>{{row['Total\_discharge']}}</td>

<td>{{row['Average\_Covered\_Charges']}}</td>

<td>{{row['Average\_Total\_Payments']}}</td>

<td>{{row['Average\_Medicare\_Payments']}}</td>

</tr>

{% endfor %}

</table>

</div>

</body>

</html>

**Output:**

**Time taken to retrieve data: 0.05 seconds**

