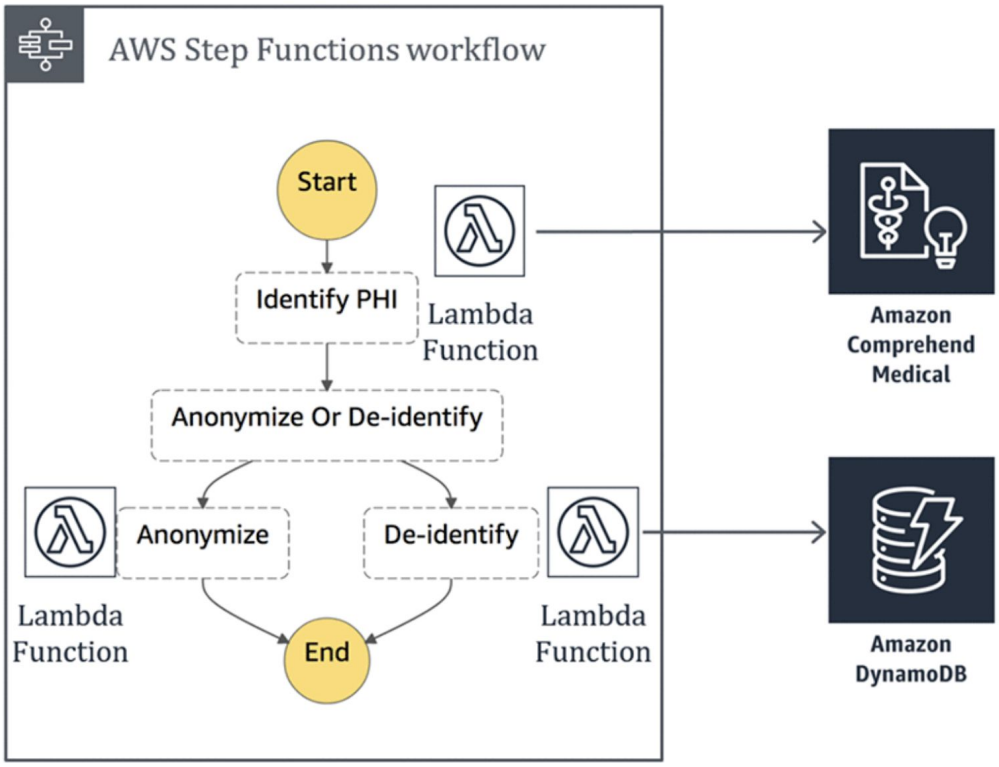
The Architecture

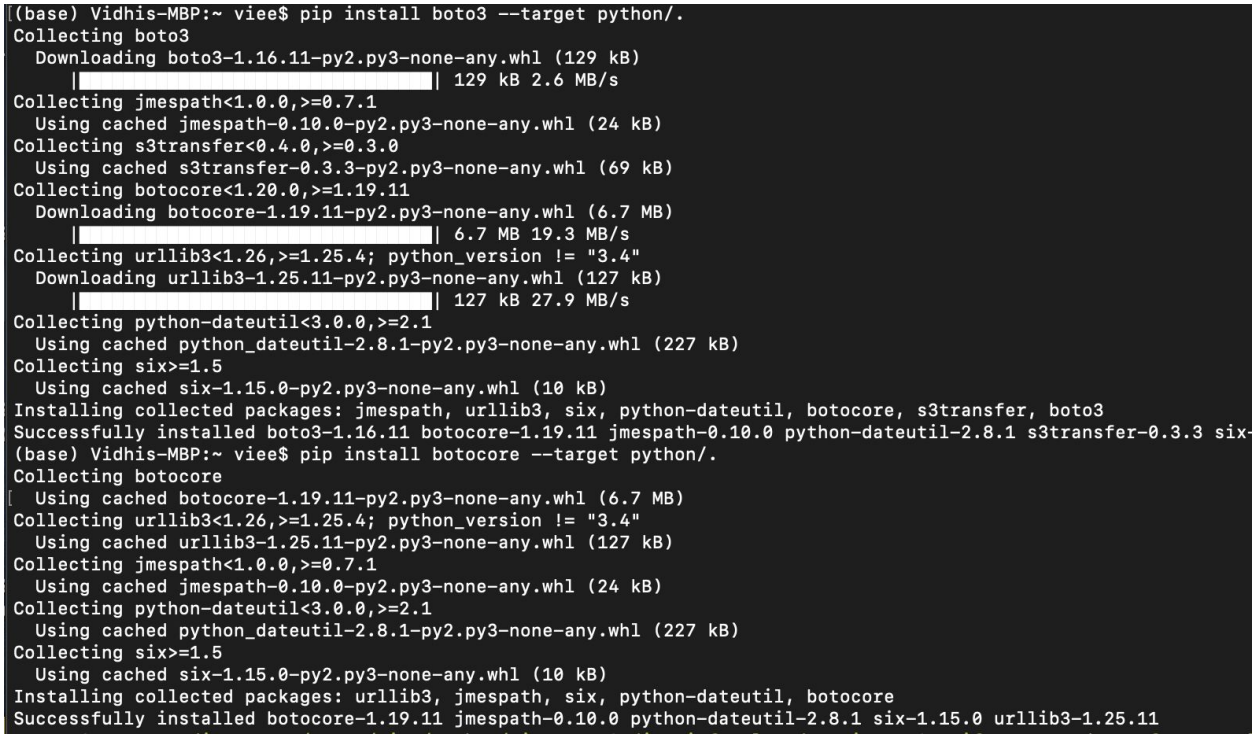


Lambda Functions

- IdentifyPHI: Uses the Amazon Comprehend Medical API to detect and identify PHI entities from a body of text, such as a medical note.
- MaskEntities: Takes the entities from IdentifyPHI as input and masks them in the body of text
- DeidentifyEntities: Takes the entities from IdentifyPHI and applies a hash to each entity and stores that mapping in DynamoDB.

Building the Boto3 Lambda layer

```
pip install boto3 --target python/.
pip install botocore --target python/.
```




```
# zip to four layer
zip boto3layer.zip -r python/
```

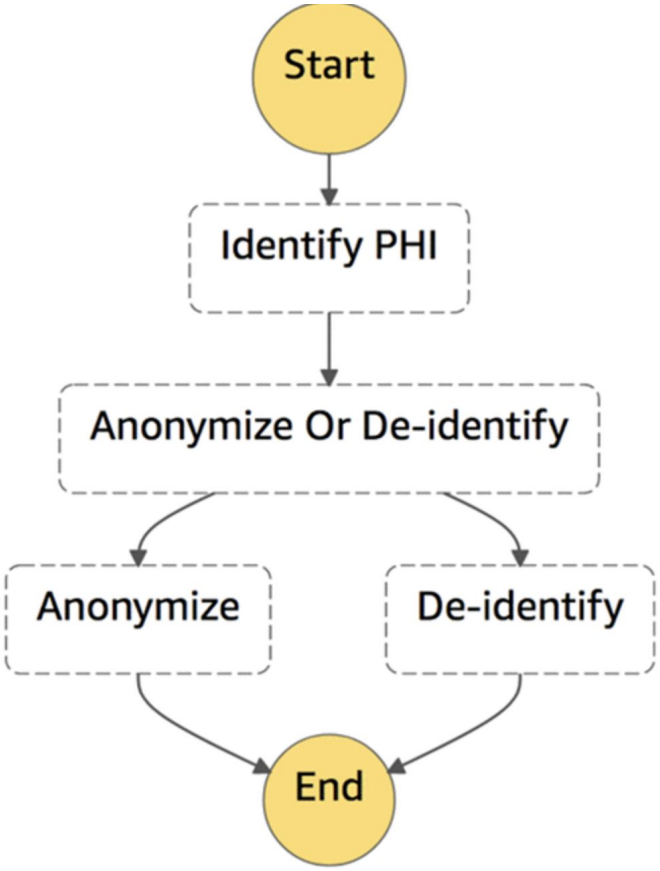
```
(base) Vidhis-MBP:~ viee$ zip boto3layer.zip -r python/
  adding: python/ (stored 0%)
  adding: python/s3transfer/ (stored 0%)
  adding: python/s3transfer/delete.py (deflated 63%)
  adding: python/s3transfer/tasks.py (deflated 72%)
  adding: python/s3transfer/compat.py (deflated 59%)
  adding: python/s3transfer/upload.py (deflated 78%)
  adding: python/s3transfer/download.py (deflated 77%)
  adding: python/s3transfer/constants.py (deflated 40%)
  adding: python/s3transfer/__init__.py (deflated 74%)
  adding: python/s3transfer/__pycache__/ (stored 0%)
  adding: python/s3transfer/__pycache__/__init__.cpython-38.pyc (deflated 58%)
  adding: python/s3transfer/__pycache__/manager.cpython-38.pyc (deflated 64%)
  adding: python/s3transfer/__pycache__/tasks.cpython-38.pyc (deflated 59%)
  adding: python/s3transfer/__pycache__/utils.cpython-38.pyc (deflated 61%)
  adding: python/s3transfer/__pycache__/processpool.cpython-38.pyc (deflated 65%)
  adding: python/s3transfer/__pycache__/upload.cpython-38.pyc (deflated 65%)
  adding: python/s3transfer/__pycache__/bandwidth.cpython-38.pyc (deflated 66%)
  adding: python/s3transfer/__pycache__/compat.cpython-38.pyc (deflated 44%)
  adding: python/s3transfer/__pycache__/futures.cpython-38.pyc (deflated 65%)
  adding: python/s3transfer/__pycache__/download.cpython-38.pyc (deflated 62%)
  adding: python/s3transfer/__pycache__/copies.cpython-38.pyc (deflated 57%)
  adding: python/s3transfer/__pycache__/exceptions.cpython-38.pyc (deflated 49%)
  adding: python/s3transfer/__pycache__/delete.cpython-38.pyc (deflated 56%)
  adding: python/s3transfer/__pycache__/subscribers.cpython-38.pyc (deflated 56%)
  adding: python/s3transfer/__pycache__/constants.cpython-38.pyc (deflated 24%)
  adding: python/s3transfer/futures.py (deflated 74%)
  adding: python/s3transfer/bandwidth.py (deflated 75%)
  adding: python/s3transfer/utils.py (deflated 73%)
  adding: python/s3transfer/processpool.py (deflated 76%)
  adding: python/s3transfer/exceptions.py (deflated 48%)
  adding: python/s3transfer/copies.py (deflated 76%)
  adding: python/s3transfer/subscribers.py (deflated 64%)
  adding: python/s3transfer/manager.py (deflated 76%)
  adding: python/six-1.15.0.dist-info/ (stored 0%)
  adding: python/six-1.15.0.dist-info/RECORD (deflated 35%)
  adding: python/six-1.15.0.dist-info/LICENSE (deflated 41%)
  adding: python/six-1.15.0.dist-info/WHEEL (deflated 14%)
  adding: python/six-1.15.0.dist-info/top_level.txt (stored 0%)
```

`aws lambda publish-layer-version --layer-name boto3-layer --zip-file fileb://boto3layer.zip`

```
(base) Vidhis-MBP:~ viee$ aws stepfunctions start-execution --state-machine-arn arn:aws:states:us-east-1:362654931460:stateMachine:DeIdandMaskSta
teMachine-khjwgWV7CssW --input file://Users/viee/BDIS/example_note.json

Error parsing parameter '--input': Unable to load paramfile file://Users/viee/BDIS/example_note.json: [Errno 2] No such file or directory: 'Users
/viee/BDIS/example_note.json'
(base) Vidhis-MBP:~ viee$ ls
Anaconda3-2020.07-MacOSX-x86_64.pkg     Java 8 Update 261.app                   data
Applications                            Library                                 hadoop-3.2.1.tar.gz
Audax                                   Movies                                  ideaIC-2020.2.2.dmg
BDIS                                    Music                                   launcher
Desktop                                 Pictures                                mongodb
Documents                               Presentation1.pptx                      mongodb-macos-x86_64-4.4.1.tgz
Downloads                               Public                                  python
EBDS                                    aws-cli                                 python-3.8.6-macosx10.9.pkg
JDK                                     boto3layer.zip
(base) Vidhis-MBP:~ viee$ cd BDIS
(base) Vidhis-MBP:BDIS viee$ ls
Assignment_1                    auth.json                       fastapi-aws-lambda-example      tutorial2_1.py
__pycache__                     cloudauth.py                    main.py                         tutorial2_2.py
archive                         example_note.json               tutorial2.py                    wildrydes-site
(base) Vidhis-MBP:BDIS viee$ aws stepfunctions start-execution --state-machine-arn arn:aws:states:us-east-1:362654931460:stateMachine:DeIdandMask
StateMachine-khjwgWV7CssW --input file://example_note.json
{
    "executionArn": "arn:aws:states:us-east-1:362654931460:execution:DeIdandMaskStateMachine-khjwgWV7CssW:3d24c10b-8341-4b12-9ff6-50a7a5291219",
    "startDate": "2020-11-05T16:41:55.460000-05:00"
}
(base) Vidhis-MBP:BDIS viee$ go version
-bash: go: command not found
(base) Vidhis-MBP:BDIS viee$ ls
Assignment_1                    auth.json                       fastapi-aws-lambda-example      tutorial2_1.py
__pycache__                     cloudauth.py                    main.py                         tutorial2_2.py
archive                         example_note.json               tutorial2.py                    wildrydes-site
```

Building the state machine



Define State Machine Code

```
{
  "Comment": "State Machine that anonymizes or deidentifies PHI",
  "StartAt": "Identify PHI",
  "States": {
    "Identify PHI": {
      "Type": "Task",
      "Resource":
"arn:aws:lambda:us-east-1:123456789012:function:IdentifyPHILambda",
      "InputPath": "$",
      "ResultPath": "$.body.entities",
      "Next": "Anonymize Or De-identify"
    },
    "Anonymize Or De-identify": {
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.body.anonymizeOrDeidentify",
          "StringEquals": "anonymize",
          "Next": "Anonymize"
```

```
        },
        {
          "Variable": "$.body.anonymizeOrDeidentify",
          "StringEquals": "deidentify",
          "Next": "De-identify"
        }
      ],
      "Default": "Anonymize"
    },
    "Anonymize": {
      "Type": "Task",
      "Resource":
"arn:aws:lambda:us-east-1:123456789012:function:MaskEntitiesLambda",
      "InputPath": "$",
      "ResultPath": "$.maskedMessage",
      "OutputPath": "$.maskedMessage",
      "End": true
    },
    "De-identify": {
      "Type": "Task",
      "Resource":
"arn:aws:lambda:us-east-1:123456789012:function:DeidentifyLambda",
      "InputPath": "$",
      "ResultPath": "$.maskedMessage",
      "OutputPath": "$.maskedMessage",
      "End": true
    }
  }
}
```

Launch Stack

https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/template?stackName=phi-detect-blog&templateURL=https://s3.amazonaws.com/aws-ml-blog/artifacts/phi-detect/phi-detect.yaml

Testing the state machine

Example_note.json

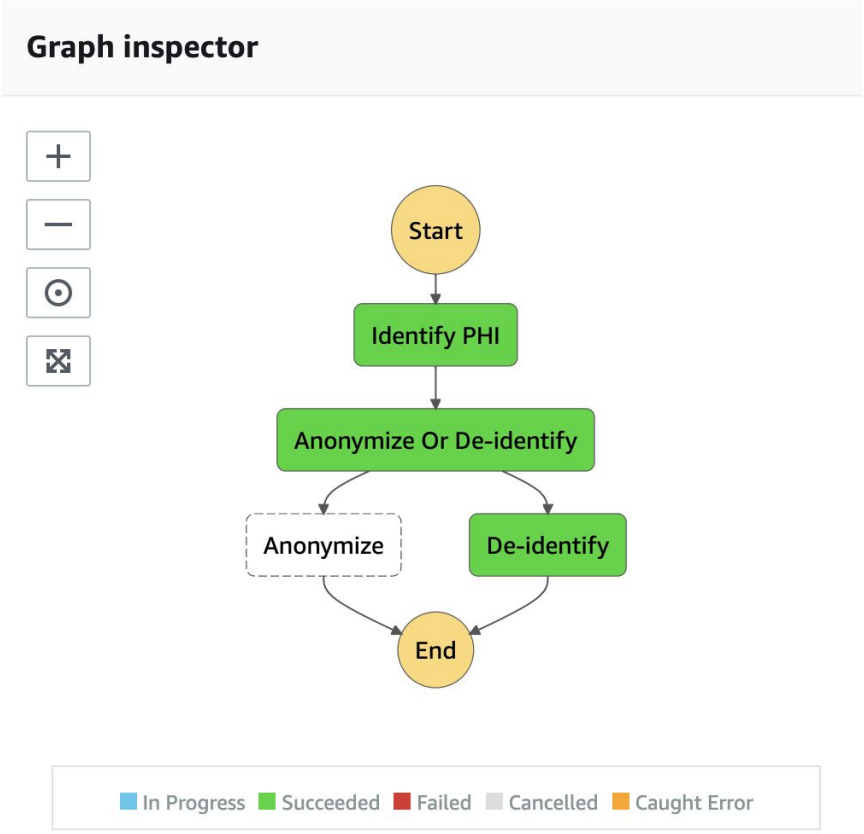
```
{
      "body": {
```

"message": " Stay Free Medical Center \nEmergency Department \nClinical Summary \n12341 W. Bohannon Rd, Grantville, GA\nPhone: (770) 922-9800 \n\n\nPERSON INFORMATION\nName: SALAZAR, CARLOS\nMRN: RQ36114734 \nED Arrival Time: 11/12/2011 18:15\n \nSex: Male \nDOB: 2/11/1961\n Age: 50 Years \nVisit Reason: New onset A Fib, SOB\n Acuity: 2 Emergent Disposition: Home/Self-Care \nAddress: 186 VALETINE, NE 69201\nPhone: 402 213-2221 \n \nSUBJECTIVE:\nCarlos came to the ED via ambulance accompanied by son, Jorge. He is a 50 yo male who was working at Food Corp when he had sudden onset of palpitations. Carlos stated his fater, Diego, also had palpitations through his life.\n \nProvider Contact Time: 11/12/2011 19:00\n Decision to Admit: Not entered\n ED Departure Time: 11/23/2011 00:07\n \nDIAGNOSIS: Hyperthyroidism \n Attending Provider: \nSaanvi Sarkar, MD\n \n Primary Nurse(s): \nJackson; Mateo\n \n\n Fill New Prescriptions:\nnepafenac (nepafenac 1 mg / 1mL Ophthalmic Suspension) 1 drop left eye every 12 hours 14 day(s)\nzofran (Ondansetron 4 mg oral tablet) 4 mg ORAL PRN\natropine sulfate 0.05 mcg / hyopscyamine sulfate 3.1 mcg / phenobartbital 48.6 MG / scopolamine hydrobromide 0.0195 mg (Donnata ER oral tablet) 1 table PO PRN\nacetaminophen - hydrocodone (Vicodin 5 mg - 500 mg oral tablet) 2 tablet(s) by Mouth every 6 hours as needed for pain\ndocusate sodium 100 mg oral capsule 100 mg by Mouth twice daily as needed for constipation\n\n \nAllergies:\n penicillins\n ibuprofen\n bee pollen\n \nPatient Education and Follow-up Information\n Instructions:\n ED, Nausea (Custom) \n Follow up:\n \n With:\nAddress:\nWhen:\n\nReturn to Emergency Department\n\n\n\nComments:\n\nNausea Vomiting\n\nNausea persists without control from anti-nausea medications Projectile vomiting Uncontrolled , consistent nausea & vomiting Blood or “coffee grounds” appearing material in vomit Medicine not kept down because of vomiting Weakness or dizziness along with nausea/vomiting Severe stomach pain while vomiting\n\nPain \nSevere Chest / Arm pain Severe squeezing or pressure in chest Severe sudden headache\nNew or uncontrolled pain New headache Chest discomfort Pounding heart Heart “flip - flop” feeling Painful Central Line site or area of “tunnel” Burning in chest or stomach Pain or burning while urinating Pain with infusion of medications or fluids into Central Line\n\n\nDiarrhea \n\nConstant or uncontrolled diarrhea New onset diarrhea Diarrhea with fever and abdominal cramping Whole pills passed in stool Greater than 5 times each day Stool which is bloody , burgundy or black Abdominal cramping\n\nFatigue\nUnable to wake\nDizziness Fatigue is getting worse Too tired to get out of bed or walk to the bathroom Staying in bed all day\n\nFever / Chills \n\nShaking chills , temperature may be normal Temperature greater than 38.3° C or 100.9° F by mouth Fever greater than 1 degree above usual if on steroids 24 Cold symptoms (runny nose , watery eyes , sneezing , coughing)
\n\n\n\nWith:\nAddress:\nWhen:\n\nFollow up with primary care provider\n\n\n\nComments:\n\nCall tomorrow to make an appointment for the

```
next 1-2 days and to start arranging PCP follow-up\n\n\nThank you for
visiting the Stay Free Medical Center.\n \n",
            "anonymizeOrDeidentify": "deidentify"
      }
}
```

Execute The following command

```
aws stepfunctions start-execution --state-machine-arn
YOUR_STATEMACHINE_ARN --input file://example_note.json
```



DeidentificationTable Close

Overview | Items | Metrics | Alarms | Capacity | Indexes | Global Tables | Backups | Contributor Insights | Triggers | Access control | Tags

Create item | Actions

Scan: [Table] DeidentificationTable: MessageHash, Entity...

Viewing 1 to 27 items

Scan | [Table] DeidentificationTable: MessageHash, EntityHash

Add filter

Start search

MessageHash	EntityHash	Entity
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	0678ce0d72b1ec41ef88d9cbdb4a4781998e039f56c1bc9d7b4757838bc7...	Emergency Department
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	0a440025854b8606d42b056d1403ca446c5f10fe7040eaffae760901efb714a9	ED
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	0dde337d96758824f7d7e4ee671831c6bb675d0aec9b336cf815ae2a82db...	Stay Free Medical Center
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	145c8df0bec6dc7116da18104b8cf679467ffba123c999b4d3953987636aa...	ED
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	1e48f50ffc971a0739b5404a8c4577e81b5e03999a262f8c31db9d434be9f5...	RQ36114734
1faa088b68aeff33fdf19c13b0b864206f67fb456819345639397981fa672d2c	2a27d4649a3c1492b379bf261fbd08ec1bbd17da05b4dbafcfc8d0b9a3c1c...	11/12/2011

Assignment 2 Documentation

Summary	In this codelab, you'll explain the tutorials.
URL	your-first-pwapp
Category	Web
Environment	web, kiosk, io2016, pwa-dev-summit, pwa-roadshow, chrome-dev-summit-2016, io2017, typtwd17, gdd17, cds17, io2018, tag-web, jsconfeu, devfest18, io2019
Status	Published
Feedback Link	https://github.com/googlecodelabs/your-first-pwapp/issues
Author	Aishwarya
Author LDAP	
Analytics Account	UA-52746336-1

Activity 1

Duration: 2:00

We will first see how we write small blocks of code in the jupyter notebooks and run a simple program.

[8]:
```
# Step 1: load modules
import numpy as np
import matplotlib.pyplot as plt
```

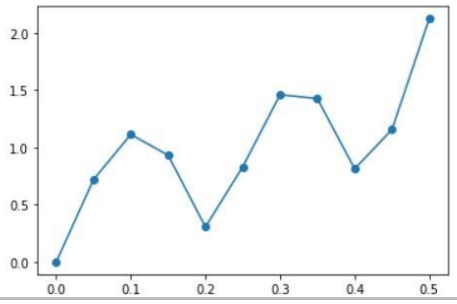
[9]:
```
# Step 2: Define domain x
x = np.linspace(0.0, 0.5, 11, endpoint=True)
print("x: ", x)
```

```
x:  [0.   0.05 0.1  0.15 0.2  0.25 0.3  0.35 0.4  0.45 0.5 ]
```

[10]:
```
# Step 3: Denote y as a function of x
y = np.sqrt( (x*x + np.sin(15*x)*np.sin(15*x)) ) / (1-x)
print("y: ", y)
```

```
y:  [0.         0.71944223 1.11388335 0.9322354  0.30596889 0.83179307
 1.46075542 1.42693321 0.81321212 1.15714049 2.12588236]
```

[11]:
```
# Step 4: Plot y
plt.plot(x, y, '-o')
plt.show()
```

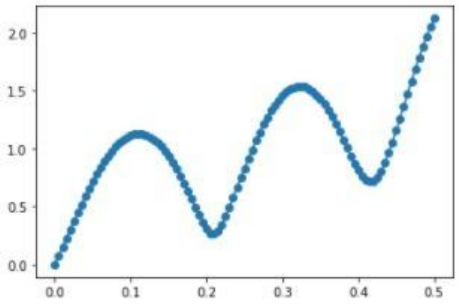


After observing the output incase we need any modifications we could make changes in the desired blocks and re-run only the necessary blocks and not the code from start to finish.

[24]:
```
# Step 3: Denote y as a function of x
y = np.sqrt( (x*x + np.sin(15*x)*np.sin(15*x)) ) / (1-x)
print("y: ", y)
```

```
y:  [0.         0.07547371 0.1512852  0.22701527 0.30224103 0.37653818
 0.44948337 0.52065653 0.58964326 0.65603716 0.71944223 0.77947513
 0.83576749 0.88796814 0.93574527 0.97878851 1.016811   1.04955126
 1.07677509 1.0982773  1.11388335 1.12345095 1.12687156 1.12407181
 1.11501492 1.09970218 1.07817449 1.05051419 1.01684728 0.97734645
 0.9322354  0.8817952  0.82637444 0.76640557 0.7024325  0.63515876
 0.56553497 0.49492403 0.42542215 0.36048046 0.30596889 0.27110648
 0.26603572 0.29372843 0.34706363 0.41636018 0.49459458 0.57748016
 0.66238513 0.74757275 0.83179307 0.91407326 0.9936084  1.06970313
 1.14173959 1.20915963 1.27145501 1.32816231 1.37886052 1.42317049
 1.46075542 1.49132213 1.51462288 1.53045762 1.53867664 1.53918362
 1.53193929 1.51696576 1.4943518  1.46425954 1.42693321 1.38271067
 1.33203923 1.27549748 1.21382602 1.14797093 1.07914504 1.0089132
 0.9393054  0.87295269 0.81321212 0.76418595 0.73045478 0.71633255
 0.72471797 0.75613439 0.80867484 0.87886122 0.96272695 1.05654868
 1.15714049 1.26187696 1.36860818 1.47555499 1.58121854 1.68431138
 1.7837082  1.87841171 1.96752976 2.05026036 2.12588236]
```

[25]:
```
# Step 4: Plot y
plt.plot(x, y, '-o')
plt.show()
```



[22]:
```
# Step 5: Change domain x
x = np.linspace(0.0, 0.5, 101, endpoint=True)
print("x: ", x)
```

```
x:  [0.    0.005 0.01  0.015 0.02  0.025 0.03  0.035 0.04  0.045 0.05  0.055
 0.06  0.065 0.07  0.075 0.08  0.085 0.09  0.095 0.1   0.105 0.11  0.115
 0.12  0.125 0.13  0.135 0.14  0.145 0.15  0.155 0.16  0.165 0.17  0.175
 0.18  0.185 0.19  0.195 0.2   0.205 0.21  0.215 0.22  0.225 0.23  0.235
 0.24  0.245 0.25  0.255 0.26  0.265 0.27  0.275 0.28  0.285 0.29  0.295
 0.3   0.305 0.31  0.315 0.32  0.325 0.33  0.335 0.34  0.345 0.35  0.355
 0.36  0.365 0.37  0.375 0.38  0.385 0.39  0.395 0.4   0.405 0.41  0.415
 0.42  0.425 0.43  0.435 0.44  0.445 0.45  0.455 0.46  0.465 0.47  0.475
 0.48  0.485 0.49  0.495 0.5  ]
```

Advantages: You can break your code into different cells and have immediate outputs. It's good for validating ideas and variable values.
You can see your changes in real-time.
You can go back to a cell and modify variables/functions at any time instead of rerunning the entire notebook.

Drawback: We could forget to change a corresponding variable/function, run other cells, or which variable you've modified before.

Activity 2

Duration: 2:00

After going through the two notebooks [i.e Notebook1 and Notebook2] we understand as the code gets more and more complex we would prefer writing scripts than executing small blocks

Some of the key reasons to use py scripting for complex coding are as follows:

- Lightweight: The .py file is 11 KB
- Version control: .ipynb files are hard to version and track diffs
- Linear development cycle
- Used in actual dev environment

Note:- As the code becomes huge and complex its becomes difficult to keep a track of errors and handle them in notebooks.

Activity 3

Duration: 1:30

Notebook to script improves performance but that is not enough.

We can further make use of python libraries to make our code efficient.

[3]:
```
import time
import random
import numpy as np

A = 1.33
N = 1000000
B = [random.random() for _ in range(N)]

start = time.time()
dist = []
for b in B:
  d = abs(A-b)
  dist.append(d)
minDist = min(dist)
end = time.time()

print('The smallest distance is ', minDist)
print('Time elapsed: ', end - start, ' s.')
```

```
The smallest distance is  0.3300000949087559
Time elapsed:  0.22688603401184082  s.
```

```
C = np.array(B) # convert to a numpy array

start = time.time()
minDist = np.min(np.abs(C-A)) # use "vectorized" numpy functions
end = time.time()

print('The smallest distance is ', minDist)
print('Time elapsed: ', end - start, ' s.')
```

```
The smallest distance is  0.3300000949087559
Time elapsed:  0.011965513229370117  s.
```

Tip: Also as seen in this example code--the numpy array functions instead of simple array helps reduce execution time sa observed in this tutorial

---

Activity 4

We can make use of profilers to get detailed runtime information on each function- this helps us narrow down on which part of the code is to be modified to improve the performance

```
(python_performance_env) D:\Parab_Ash\MIS\BD_pipeline\Ass2P1\2020-ComputeFest\notebook_to_cloud\PythonPerformance\regenerative_morph>python regenerative_morph_slow.py

Creating Frame 55
         174801200 function calls in 943.906 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
  9199428   20.156    0.000   20.156    0.000 :0(append)
       18    0.000    0.000    0.000    0.000 :0(array)
        1    0.000    0.000  943.906  943.906 :0(exec)
 18398857   77.688    0.000  640.750    0.000 :0(implement_array_function)
     3992    0.266    0.000    0.266    0.000 :0(index)
 18398857   38.266    0.000   38.266    0.000 :0(isinstance)
 18398857   38.688    0.000   38.688    0.000 :0(items)
     3992    0.562    0.000    0.562    0.000 :0(min)
 18398856  132.812    0.000  132.812    0.000 :0(reduce)
        9    0.000    0.000    0.000    0.000 :0(reshape)
        3    0.000    0.000    0.000    0.000 :0(zeros)
 18398857  104.234    0.000  778.016    0.000 <__array_function__ internals>:2(sum)
        1    0.000    0.000  943.906  943.906 <string>:1(<module>)
        9    0.000    0.000    0.000    0.000 _asarray.py:14(asarray)
 18398857   33.031    0.000   33.031    0.000 fromnumeric.py:2100(_sum_dispatcher)
 18398857  137.406    0.000  563.062    0.000 fromnumeric.py:2105(sum)
 18398857  172.969    0.000  387.391    0.000 fromnumeric.py:70(_wrapreduction)
 18398857   42.922    0.000   42.922    0.000 fromnumeric.py:71(<dictcomp>)
        1    0.000    0.000  943.906  943.906 profile:0(morph = regenerative_morph(source1, source2, target, ws, 0.5))
        0    0.000             0.000          profile:0(profiler)
        1    0.562    0.562  943.906  943.906 regenerative_morph_slow.py:136(target2sources)
        9    0.000    0.000    0.000    0.000 regenerative_morph_slow.py:19(im2col_sliding_strided)
        1    0.000    0.000  943.906  943.906 regenerative_morph_slow.py:274(regenerative_morph)
        3    0.000    0.000    0.000    0.000 regenerative_morph_slow.py:36(create_patches)
     3993  144.344    0.036  943.344    0.236 regenerative_morph_slow.py:54(patch_match)
        9    0.000    0.000    0.000    0.000 stride_tricks.py:19(__init__)
        9    0.000    0.000    0.000    0.000 stride_tricks.py:24(_maybe_view_as_subclass)
        9    0.000    0.000    0.000    0.000 stride_tricks.py:37(as_strided)
```

Activity 5

Comparing the scripts runtime :

```
Creating Frame 55
         2416041 function calls in 109.297 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
   241000    0.500    0.000    0.500    0.000 :0(append)
      301    0.000    0.000    0.000    0.000 :0(array)
        2    0.000    0.000    0.000    0.000 :0(copy)
        1    0.000    0.000  109.297  109.297 :0(exec)
   157730    0.719    0.000    8.562    0.000 :0(implement_array_function)
   139115    6.094    0.000    6.094    0.000 :0(index)
   157730    0.391    0.000    0.391    0.000 :0(isinstance)
   157730    0.234    0.000    0.234    0.000 :0(items)
   120500    0.344    0.000    0.344    0.000 :0(keys)
    18615    6.609    0.000    6.609    0.000 :0(min)
   157730    4.031    0.000    4.031    0.000 :0(reduce)
       27    0.000    0.000    0.000    0.000 :0(reshape)
        1    0.000    0.000    0.000    0.000 :0(setprofile)
   120500    0.281    0.000    0.281    0.000 :0(values)
    81819    1.859    0.000    1.859    0.000 :0(zeros)
   157730    1.047    0.000    9.984    0.000 <__array_function__ internals>:2(sum)
        1    0.000    0.000  109.297  109.297 <string>:1(<module>)
       27    0.016    0.001    0.016    0.001 _asarray.py:14(asarray)
   157730    0.375    0.000    0.375    0.000 fromnumeric.py:2100(_sum_dispatcher)
   157730    1.125    0.000    7.844    0.000 fromnumeric.py:2105(sum)
   157730    1.703    0.000    6.328    0.000 fromnumeric.py:70(_wrapreduction)
   157730    0.359    0.000    0.359    0.000 fromnumeric.py:71(<dictcomp>)
        1    0.000    0.000  109.297  109.297 profile:0(morph = regenerative_morph(source1, source2, target, ws, 0.5))
        0    0.000             0.000          profile:0(profiler)
```

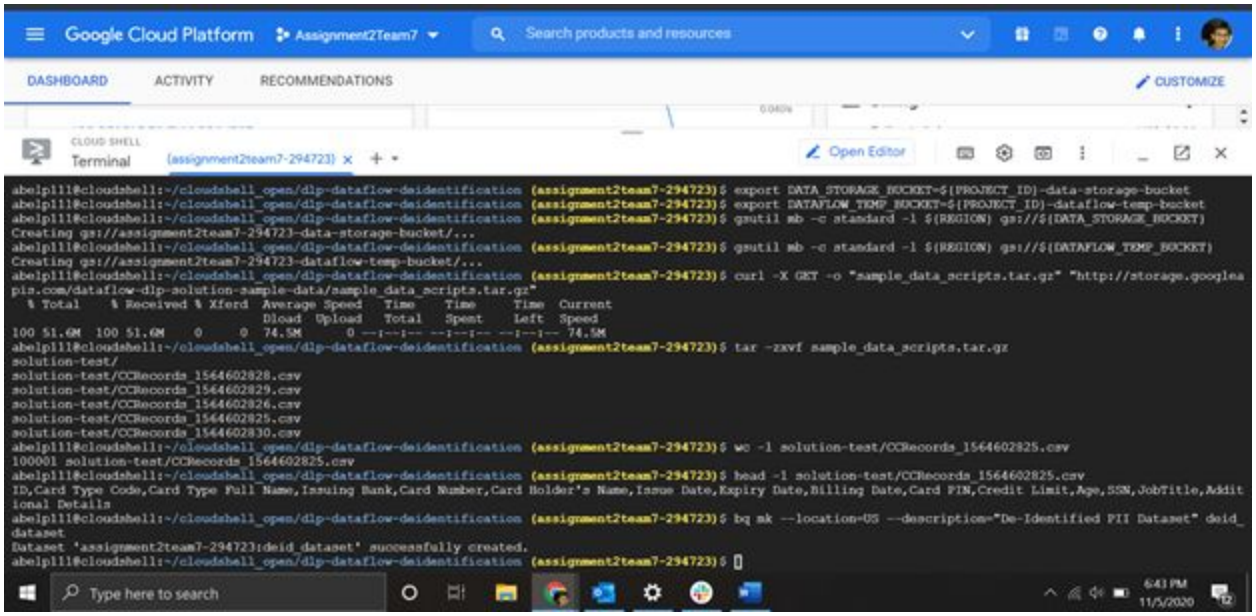
Conclusion:
There are pros and cons to using notebooks and switching from notebook to scripting does

not solve all the problems.

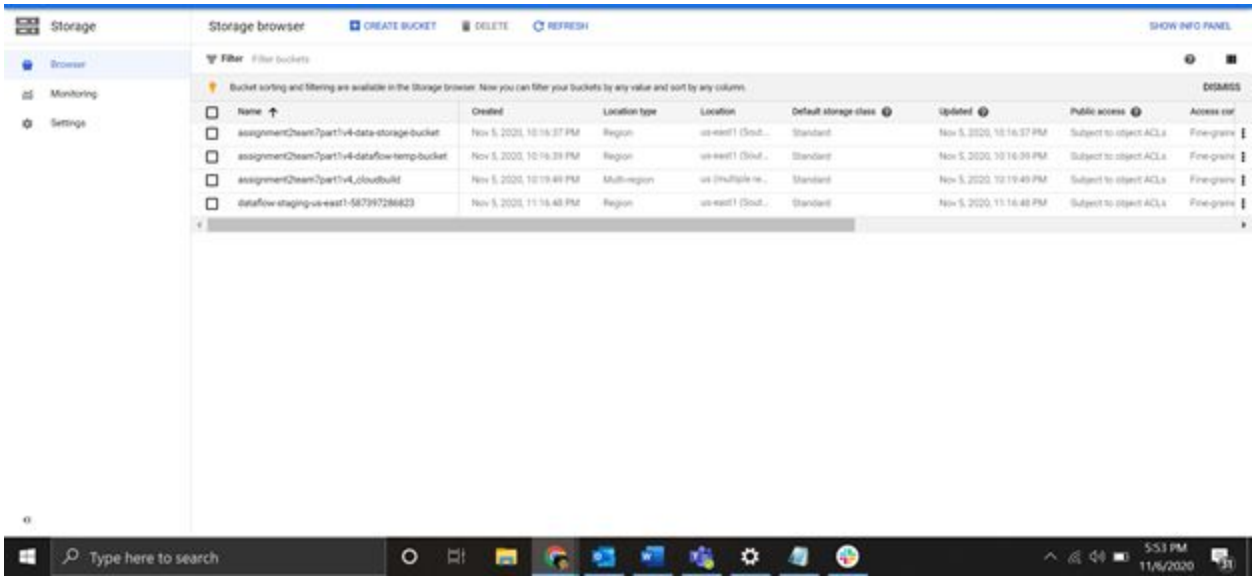
For complex codes like transfering images it would take 2-3 hours on a CPU to generate a result, whereas on AWS EC2 GPU it takes a couple minutes.

De-identifying and Re-identifying Data

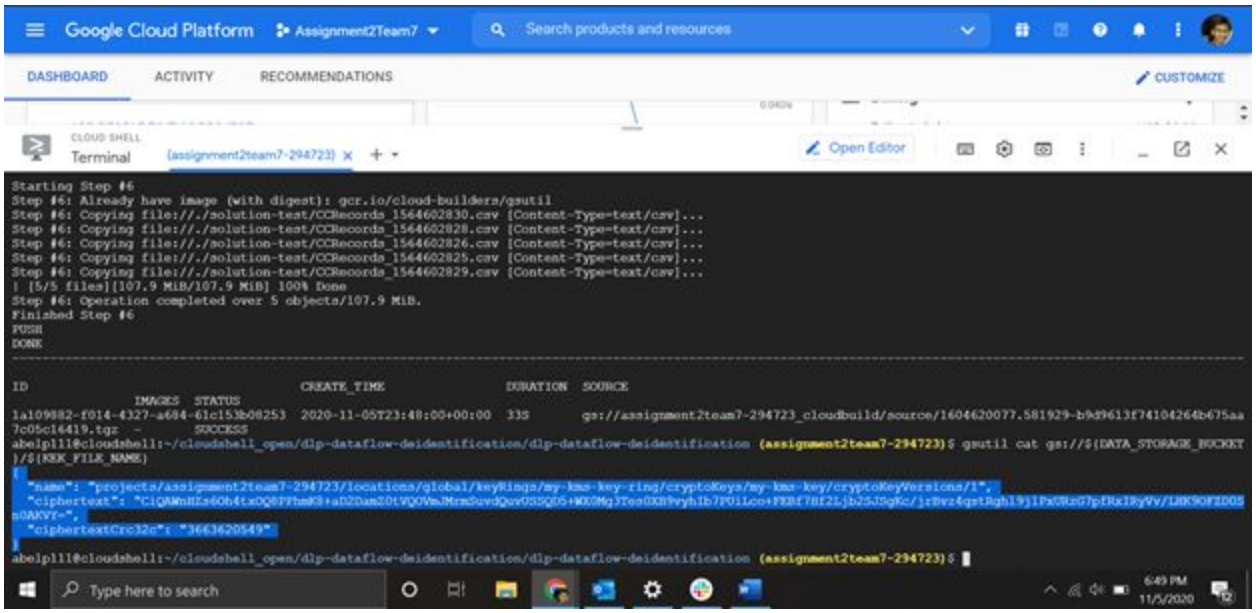
Configured services, downloaded datasets and Created buckets and BigQuery



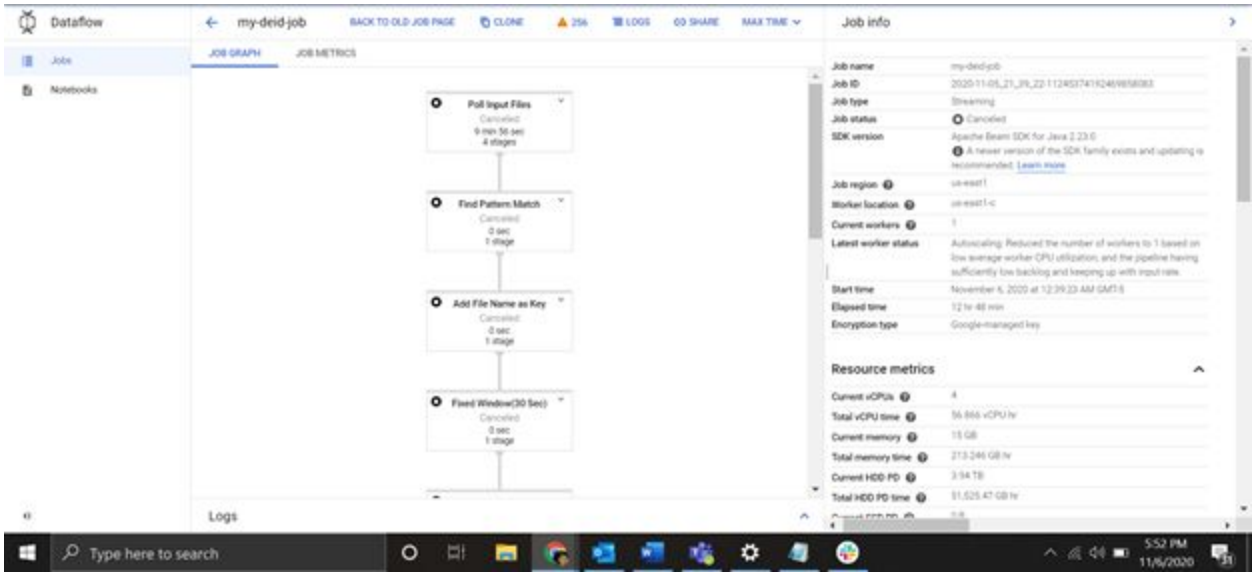
Created Buckets -



Created and configured keys



Dataflow to de-identify data-



Re-identifying DataFlow -