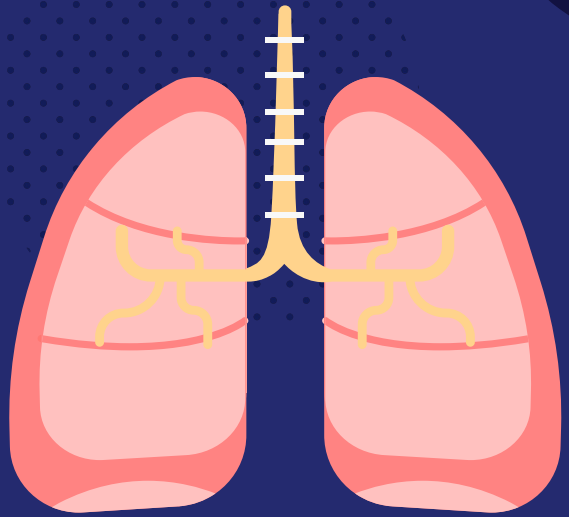


HEALTHCARE ANALYSIS: RESPIRATORY ILLNESS

PRESENTED BY THE S3 AMIGOS

JASMINE JIANG | ETTIONE STUCKEY II | JUSTIN QUINN

TABLE OF CONTENTS

YIKES!

Respiratory illness is an ever-growing problem, threatening the lives of Americans all across the country ...

YAY!

The good news is, The S3 Amigos are on the case! With our skills in data analytics and AWS, we can recognize trends and patterns in the data and provide a basis on which future healthcare strategies will be developed!

TEAM STRATEGY:
DIVIDE & CONQUER

OBJECTIVES

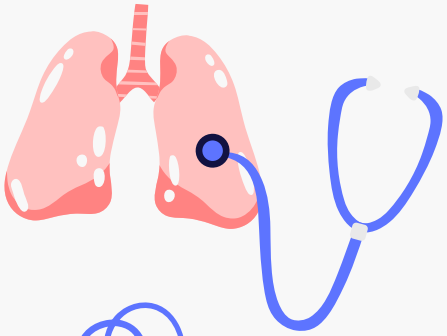
- Create a working pipeline from file upload to cloud RDS
- Clean data via deployed Lambda function
- Upload data to RDS
- Connect locally to RDS to generate analytics
- Display analytics results on EC2-hosted website

02

SCHEMA

ER DIAGRAM

Outpatient_Respiratory_Illness_Activity	
PK	entry_id INTEGER NOT NULL
	week_ending DATE
	week INTEGER
	season TEXT
	state TEXT
	activity_level TEXT
	activity_level_label TEXT



AWS ARCHITECTURE

Let’s take a look at how our AWS services are set up and connected to each other.

03

CHECKPOINT 1: SETTING UP THE S3

CONFIGURATION: S3

Select a region

AWS Region

US East (N. Virginia) us-east-1

Create a name

Bucket name Info

myNewBucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

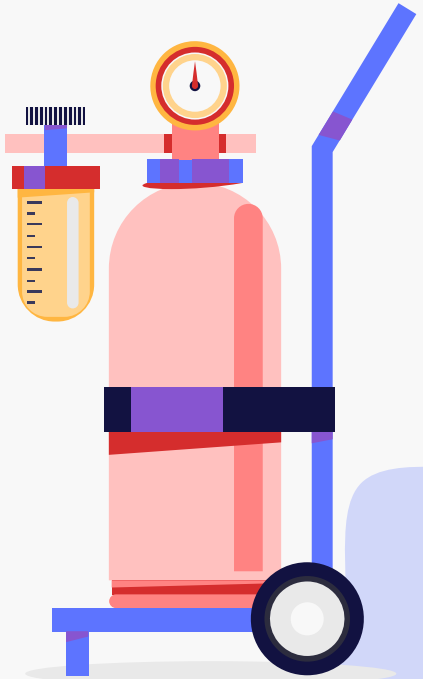
Block public access

Block *all* public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Create bucket

That's it! The S3 has been created.



04

CHECKPOINT 2: SETTING UP THE RDS

CONFIGURATION: RDS

Select an engine & template

MySQL

Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

Set db identifier & credentials

DB instance identifier Info
Type a name for your DB instance. Region.
database-1
The DB instance identifier is case-ir characters or hyphens. First charact

Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. The first character mu

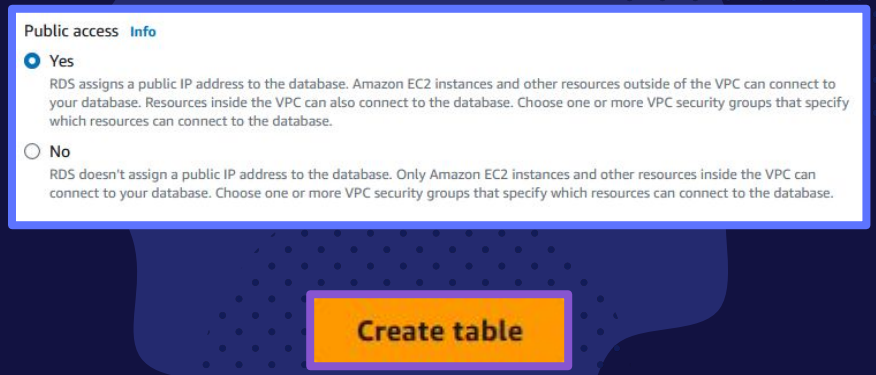
Master password Info
••••••••
Constraints: At least 8 printable ASCII characters.

Select the default VPC

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.
Default VPC (vpc-0b6e7453c3c8c29b4)
6 Subnets, 6 Availability Zones
Only VPCs with a corresponding DB subnet group are listed.

WHOA!

Remember to set your RDS public access to “yes”. Failure to do so could cause complications when trying to access the database with other services!



05

CHECKPOINT 3: LAMBDA TO RDS

CONNECTING TO DATABASE

- Created new Lambda function
- Assigned appropriate permissions and security groups
- Used Pymysql library to create connection to RDS database and to clean data
- Attached function to S3 trigger

POPULATING DATABASE

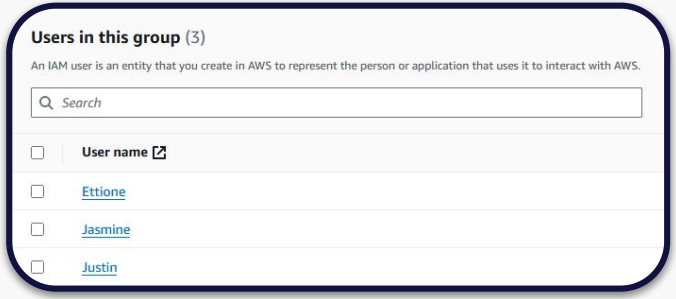
```
bucket = event['Records'][0]['s3']['bucket']['name']
key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
local_file_path = '/tmp/' + os.path.basename(key)
s3.download_file(bucket, key, local_file_path)
logger.info("SUCCESS: File retrieved from S3")

data = []
with open(local_file_path, 'r') as csvfile:
    csv_reader = csv.reader(csvfile)
    for row in csv_reader:
        data.append(row[:4] + [row[4].replace("Level ", '')] + row[5:])
    data = [tuple(row) for row in data]
logger.info("SUCCESS: File read")
data = data[1:]
```

- Downloaded .csv file from S3 bucket into /tmp folder
- Executed insert statement with parsed list of tuples
- Cleaned the extracted data

PERMISSIONS: IAM

Although our resources have now been configured, there are still a few things that need to be done before the team can access them.

- IAM users must be created for all group members.
- IAM users must be added to a group.
- A policy must be created and added to the group.

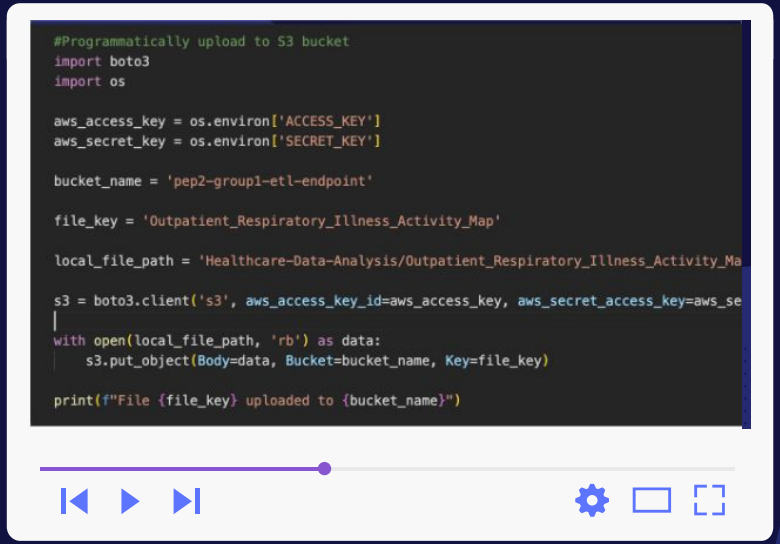


```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:*",
                "iam:*",
                "ec2:*",
                "logs:*",
                "q:*",
                "ec2-instance-connect:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "lambda:*",
                "rds:*",
                "cloudwatch:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

06

CHECKPOINT 4: RDS TO ANALYSIS

Upload to the S3 Bucket



```python
#Programmatically upload to S3 bucket
import boto3
import os

aws_access_key = os.environ['ACCESS_KEY']
aws_secret_key = os.environ['SECRET_KEY']

bucket_name = 'pep2-group1-etl-endpoint'

file_key = 'Outpatient_Respiratory_Illness_Activity_Map'

local_file_path = 'Healthcare-Data-Analysis/Outpatient_Respiratory_Illness_Activity_Ma

s3 = boto3.client('s3', aws_access_key_id=aws_access_key, aws_secret_access_key=aws_se

with open(local_file_path, 'rb') as data:
    s3.put_object(Body=data, Bucket=bucket_name, Key=file_key)

print(f"File {file_key} uploaded to {bucket_name}")
```

Be sure to keep those keys private!

Connect to RDS + Upload CSV

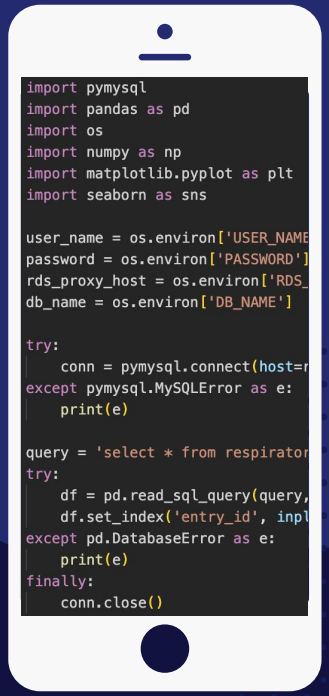
Be sure you have the correct RDS Proxy Host and DB Name.

Using Try-Except blocks is good practice!

Pull from RDS + Convert to Dataframe

Double check your queries to ensure everything runs smoothly.

Congrats! You are now ready to do some analysis!



```
import pymysql
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

user_name = os.environ['USER_NAME
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_
db_name = os.environ['DB_NAME']

try:
    conn = pymysql.connect(host=r
except pymysql.MySQLError as e:
    print(e)

query = 'select * from respirator
try:
    df = pd.read_sql_query(query,
    df.set_index('entry_id', inpl
except pd.DatabaseError as e:
    print(e)
finally:
    conn.close()
```

07

EXTENSION: HOSTING ON EC2

Creating the EC2

Step 1: Enter EC2 Dashboard

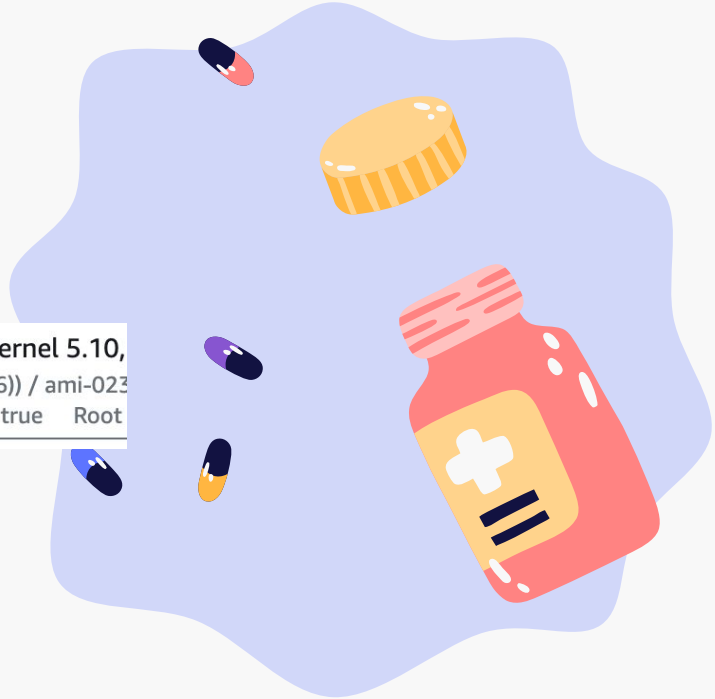
Step 2: Select ‘Instances’

Step 3: Launch an Instance

Step 4: Select OS

Step 5: AMI

Step 6: Architecture



Creating the EC2+ WebHosting

Step 7: Instance Type

Instance type

t2.micro

Family: t2 1 vCPU 1

Step 8: Key Pair

Create new key pair

Create key pair

Step 9: Launch

Launch instance

Connect

Root User

```
$ sudo su -
```

Install

```
]$ yum install -y httpd
```

Host

```
systemctl enable httpd
```

08

LESSONS LEARNED

CHALLENGE #1

The Problem:

Connection failed during testing despite configured environment variables

Tried multiple potential endpoints for host variable

Finally pinpointed and solved issue by configuring security groups for Lambda function

The Lesson:

Don't get fixated on what you may *think* is the problem, and consider potential outside factors as well

CHALLENGE #2

The Problem:

File from S3 bucket could not be downloaded via Lambda

Created a new Lambda function from scratch, which *could* download and process the file, and compared them

Only difference was that the original function was in a VPC

I unhooked the function from the VPC and set the host to the public RDS address

The Lesson:

Go with what works, even if you don't entirely understand the fine mechanics

Use process of elimination and comparison with working code to pinpoint issues

CHALLENGE #3

The Problem:

Was unable to connect to AWS RDS from local machines.

Reconfigured security groups, proxies, and VPCs to no avail.

Finally, went through our RDS configurations and noticed that public access was set to “No”.

After changing this setting to “yes” and waiting a couple minutes for the modification to take effect, we were finally able to connect to the database.

The Lesson:

Sometimes the *simplest* solution is the best solution. Be sure to carefully look over the configurations for your service before it is deployed!

CHALLENGE #4

The Problem:

Was unable to remote connect into the EC2 from a local machine

Even though the key file and EC2 address were correct, my access was denied.

The error message stated my key was not secure enough

This require that I change the permissions on the key file directly through the command line

The Lesson:

Read! Very often the system is telling you exactly what you are doing wrong. Understanding error messages can help point you in the right direction to find a solution.

DEMO TIME!

Jira Board

Website