**Final project  
   
NAYA COLLAGE  
    
08/2021**

A picture containing graphical user interface

Description automatically generated

**revital.matityahu@gmail.com 054-8050636**

**eransarid@gmail.com 052-8757584**

**Revital MATITYAHU & ERAN Sarid**

Table of Contents

[1 Introduction 3](#_Toc81130153)

[2 Project Architecture 4](#_Toc81130154)

[3 AWS Services - Used 5](#_Toc81130155)

[3.1 EC2 5](#_Toc81130156)

[3.2 Amazon S3 5](#_Toc81130157)

[3.3 Cloud9 6](#_Toc81130158)

[3.4 DynamoDB 7](#_Toc81130159)

[3.5 Amazon Kinesis Data Streams (KDS) 8](#_Toc81130160)

[3.6 Lambda 9](#_Toc81130161)

[3.7 Kinesis Firehose 10](#_Toc81130162)

[3.8 QuickSight 11](#_Toc81130163)

[4 Main Packages - Used 12](#_Toc81130164)

[4.1 Package: Boto3 12](#_Toc81130165)

[4.2 Package: flask 12](#_Toc81130166)

[5 Files and Folders - overall 13](#_Toc81130167)

[6 Deploy Flask Environment 14](#_Toc81130168)

[How to Execute Python Flask File on AWS 14](#_Toc81130169)

# Introduction

A web application for online commercial between consumers and consumers/suppliers.

It can serve both consumers and bruisedness cause its win-win app.

The business can save lot of marketing budget since he gets access to consumers leads so he can make discount offers so the consumers can get better deals and save lot time for searching offers on the net.

Each logged in consumers may add multiple bids in several categories and he can see associated offers given by other consumer (for secondhand) or related business in case of flights\hotels bids.

The consumer sees his own open/closed bids and can filter and sort by different attributes.

Some bids (according to the selecting category) such flights and hotels requires target price value.

Each consumer may also apply offer for other open bids initiated by other consumers.

Some categories have enrichment information that can assist determinate the target price for specific bid.

For instance, Flights category having enrichment related to current flights prices (scrapped from the web).

Consumer / business who intends apply offers logged in into the system, sees open bids per category and

Apply offer. In case the bid has a target price the offer price must be below that price.

When the bid initiator approved that offer the bid become closed and he can see its contact information for going FW to financial transaction form.

# Project Architecture

Timeline

Description automatically generated

# AWS Services - Used

# EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides computing environment.

Where and Why we use it?

We have two Ec2 for :

* Execute Airflow to run our program
* We use EC2 to execute python files on cloud9

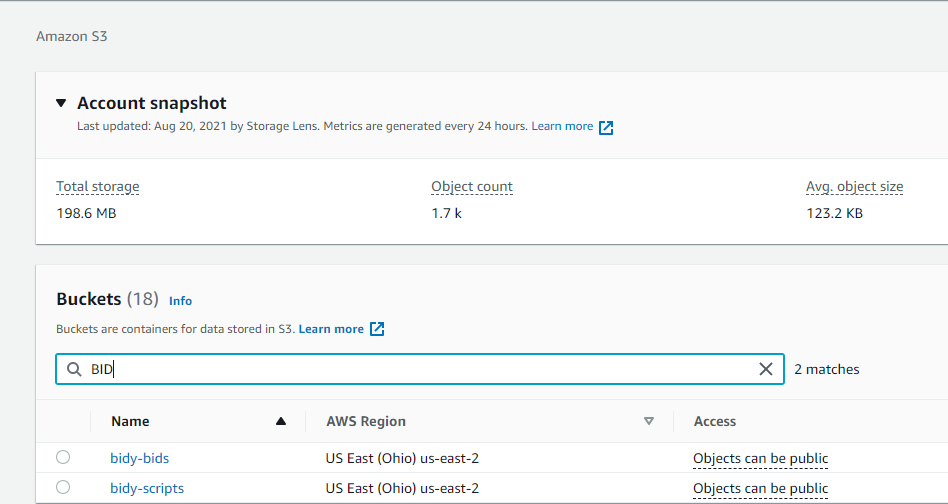
# Amazon S3

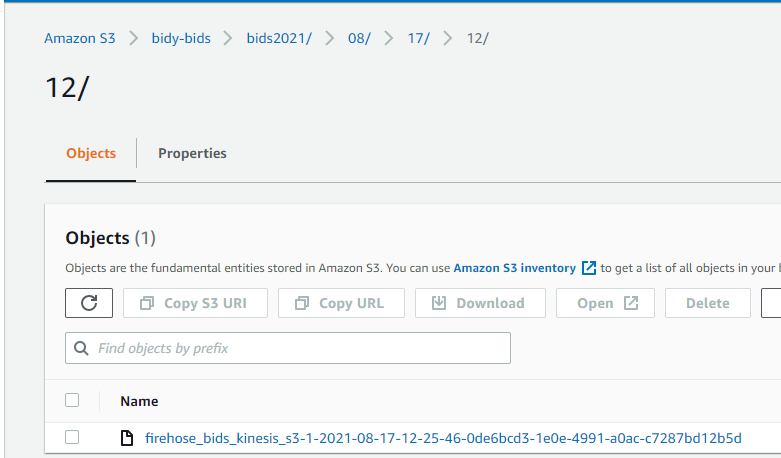
Amazon S3 is an object storage service

Where and Why we use it?

S3 offer scalability, data availability, security, and performance.

We use s3 to store json files for archive and analytics





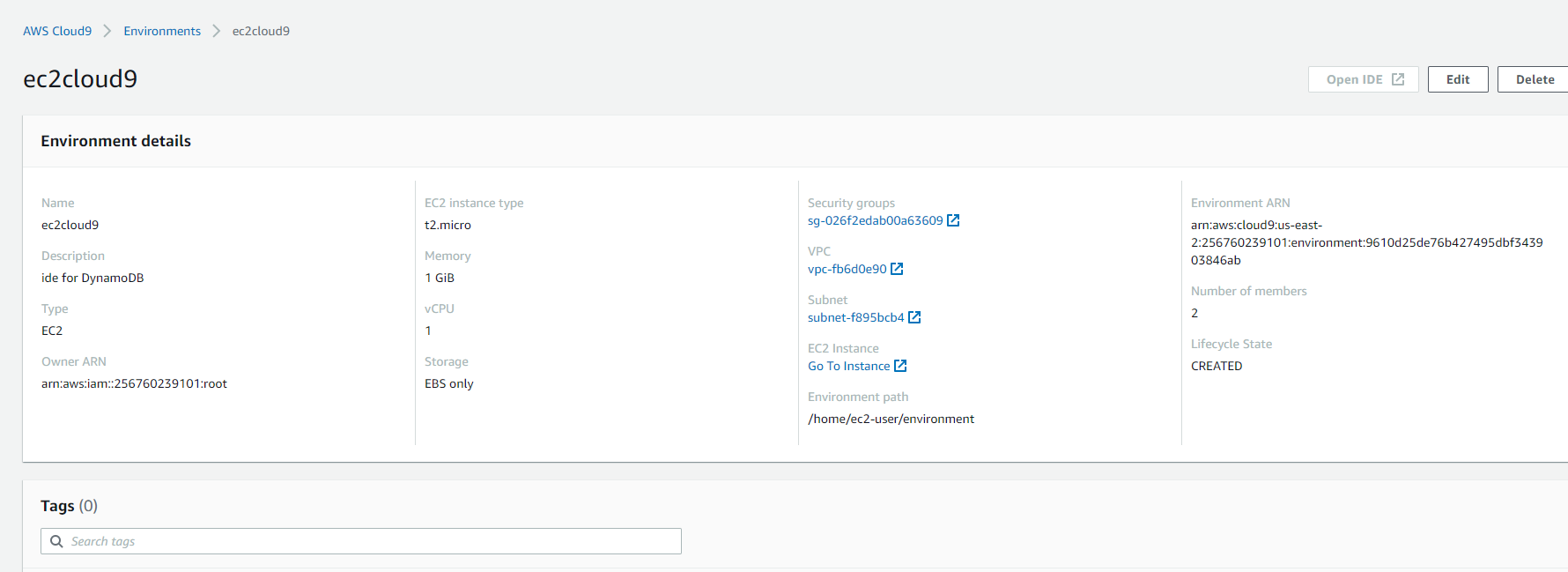
# Cloud9

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. With cloud9 we could share our development environment with my team.

Where and Why we use it?

Cloud9 is easily integrated to other aws services.

Our program was developed in that service as development environment mode.



# DynamoDB

DynamoDB is NoSQL database, as it’s fast, scalable, requires 0 maintenance, has a simple API, and is a joy to work with.

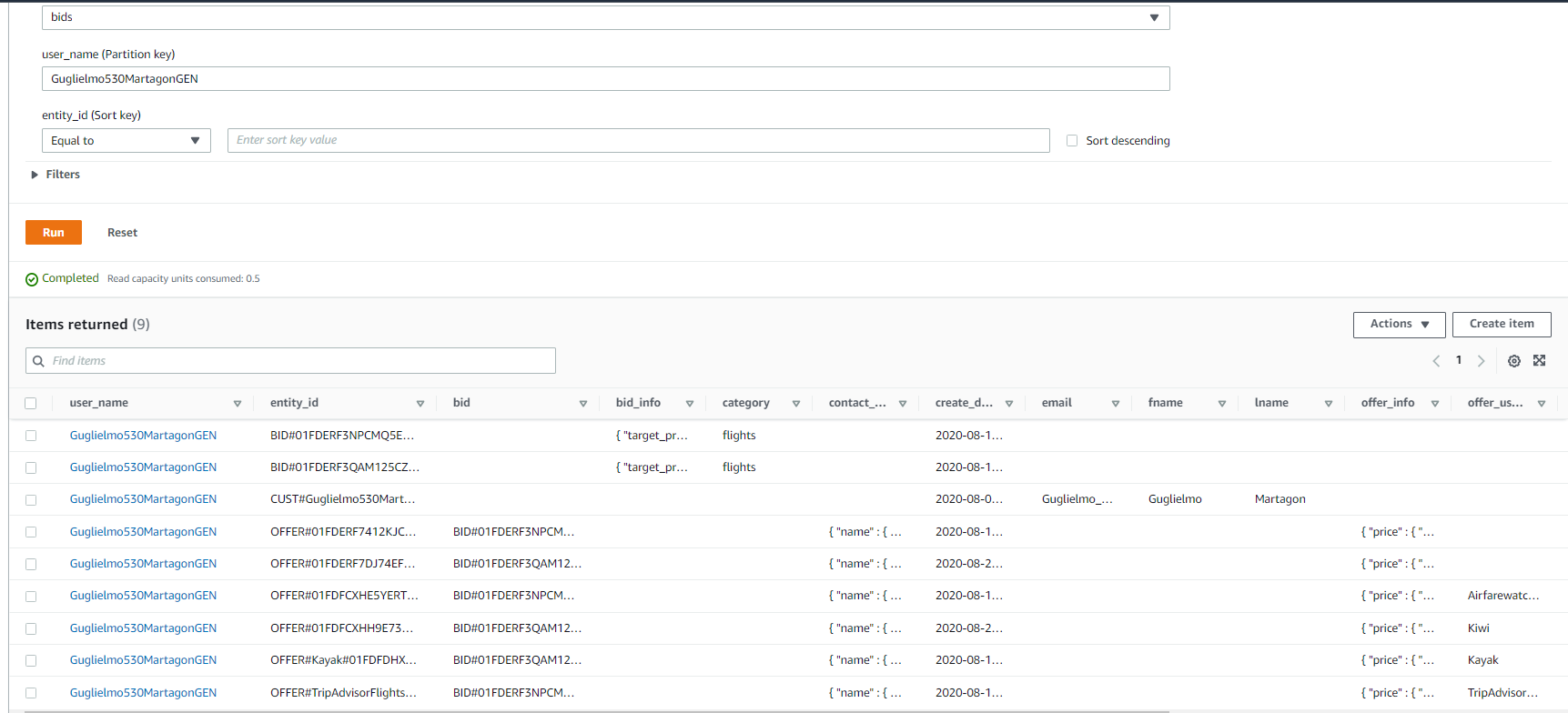
Where and Why we use it?

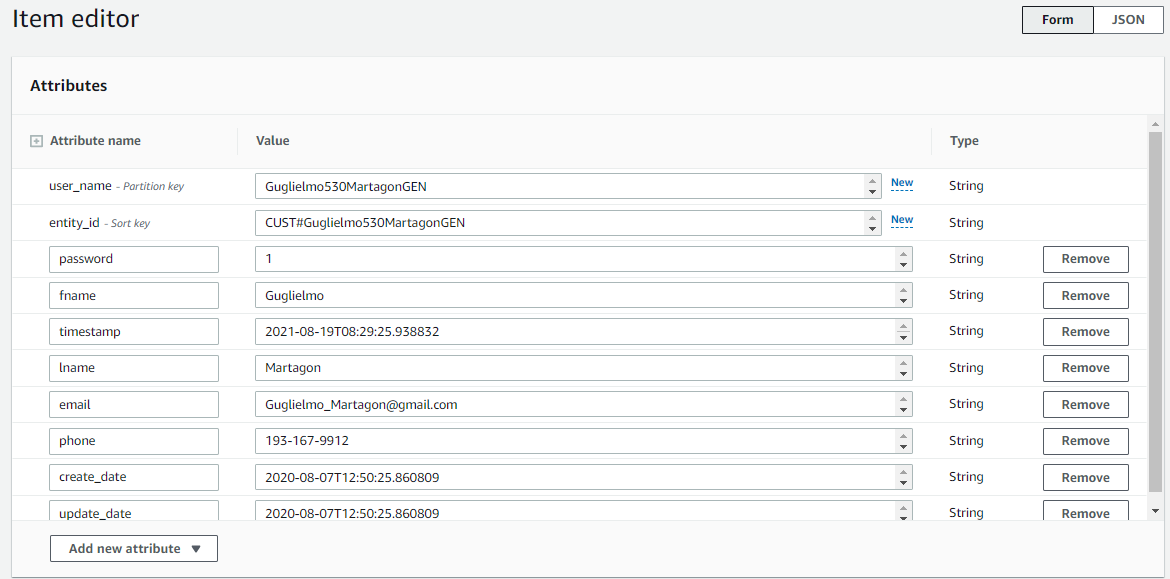
DynamoDB is the main DB for our project. It Is the operational DB for Bids and Trips meaning that we save only the last updated data. (in S3 we save the historical data)

We use it to store the following items:

Bids Table: Users \ Bids \ Offers generated from the Bids web app

R2rTrips Table: Extracted trips info and statistics scrapped from Rome2Rio web site





# Amazon Kinesis Data Streams (KDS)

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources.

Make streaming data available to multiple real-time analytics applications, to Amazon S3, or to AWS Lambda.

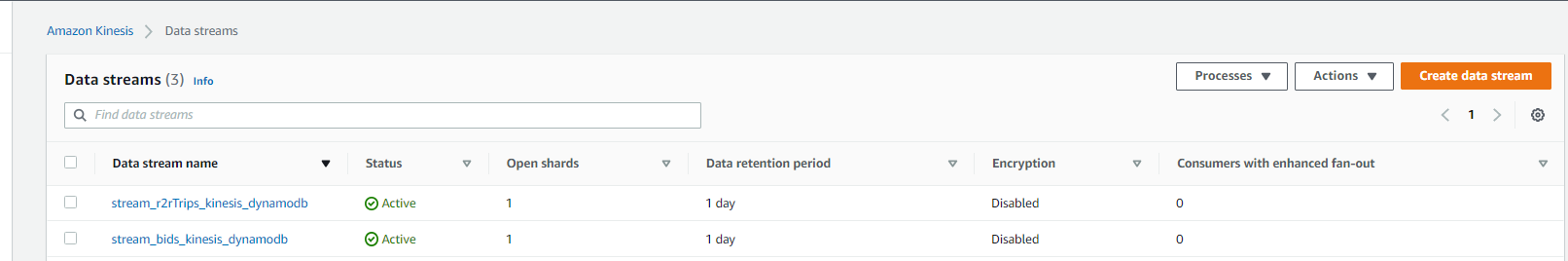
Where and Why we use it?

We use kinesis as pipeline for transfer data through Lambda to DynamoDB

and as input source for Kinesis Firehose.

There are 2 data streams:

* Bids app data
* Rome2Rrio scrapping data



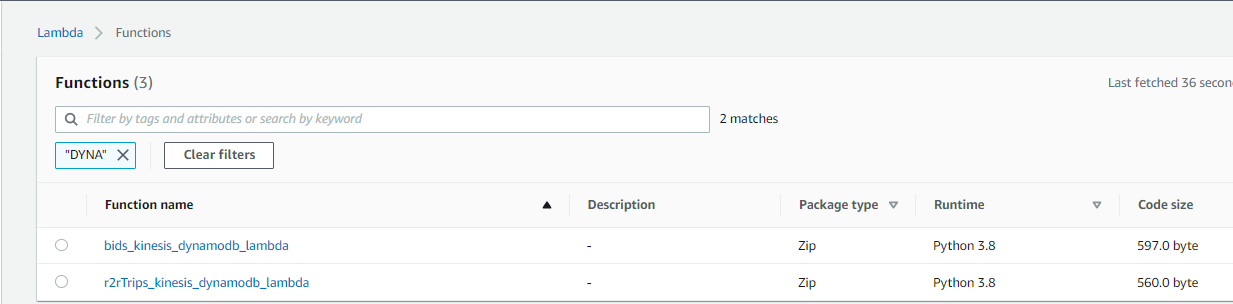
# Lambda

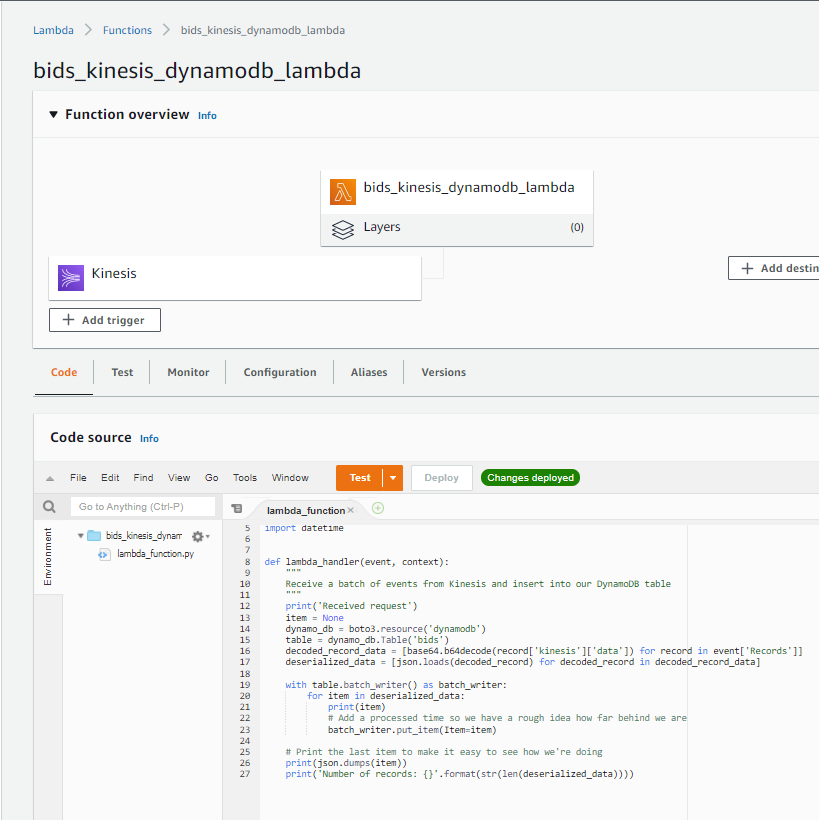
AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes.

Where and Why we use it?

We use Lambda to write bids and trips Items into DynamoDB tables.

It can use us later on for manipulating data, cleansing, add additional attributes etc.





# Kinesis Firehose

The Kinesis Firehose connector publishes data through an Amazon Kinesis Data Firehose delivery stream to destinations such as Amazon S3, Amazon Redshift, or Amazon Elasticsearch Service.

Where and Why we use it?

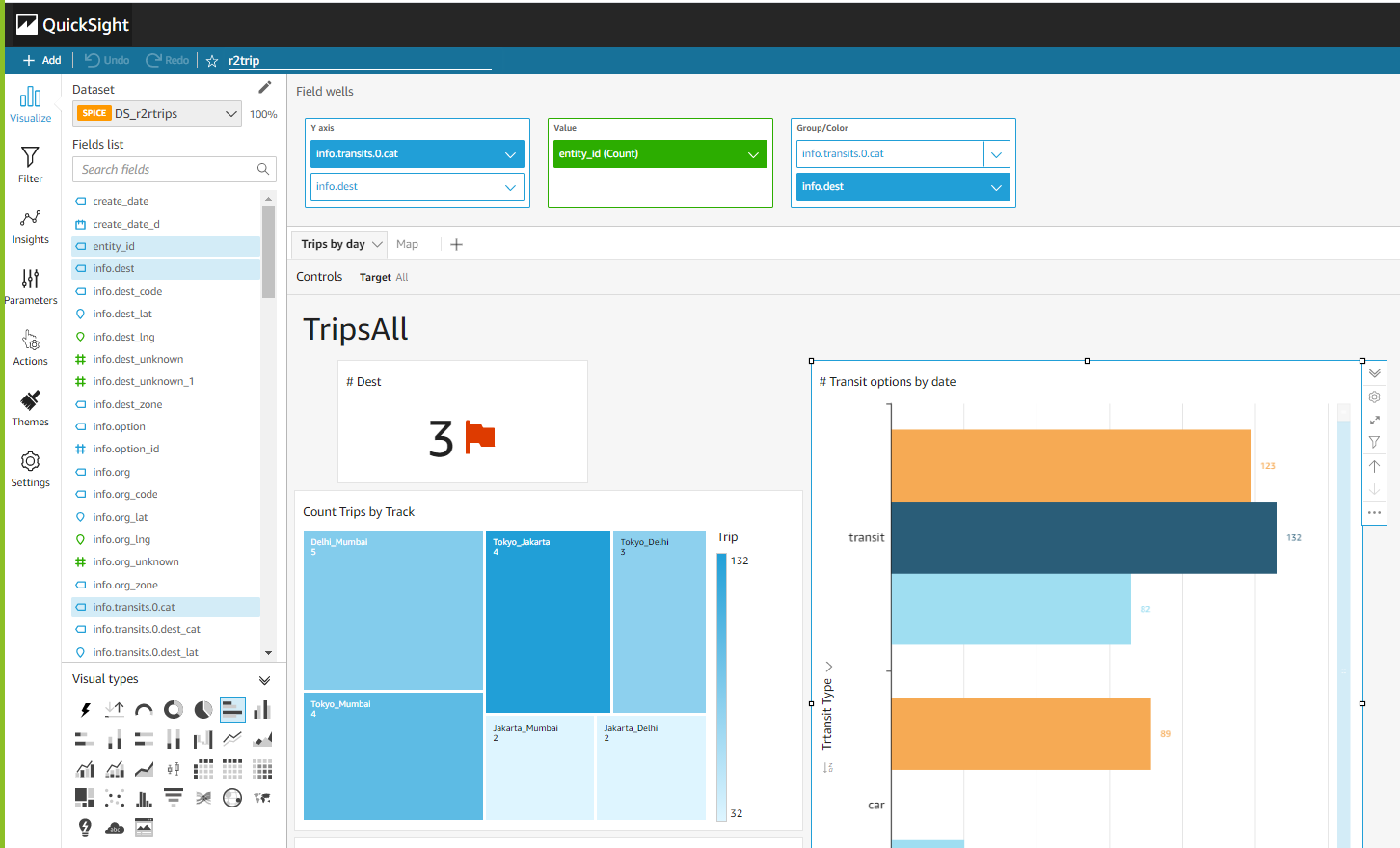
We use it in order to store a data stream events in S3 for archive and analytics.

# QuickSight

Amazon QuickSight is a scalable, serverless, embeddable, machine learning-powered business intelligence (BI) service.

Where and Why we use it?

We use it for create analytics and display Dashboards about Bids and Trips.



https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/workbench.settingup.html

# Main Packages - Used

# Package: Boto3

**Why Using:**

Boto3 is the **name of the Python SDK for AWS**. It allows you to directly create, update, and delete AWS resources from your Python scripts.

We use boto3 to connect with Python to DynamoDB Services,

**Documentation**:

* <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html>
* https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html

# Package: flask

**Why Using:**

Flask is based on Werkzeug a WSGI utility library and Jinja2 which is its template engine. You can use this **web application framework to compile modules and libraries** which will also help the developer to write web applications without writing low-level code like thread management and protocols**.**

Our application "Market Match Place" – based on flask packages.

**Documentation**:

* https://flask.palletsprojects.com/en/2.0.x/

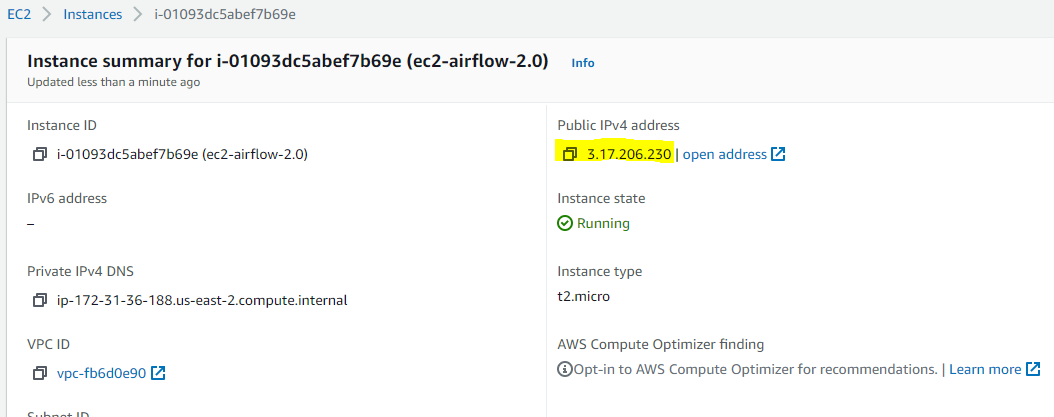
# Files and Folders - overall

**Attached in** **FilesAndFolder.xlsx**

# Deploy Flask Environment

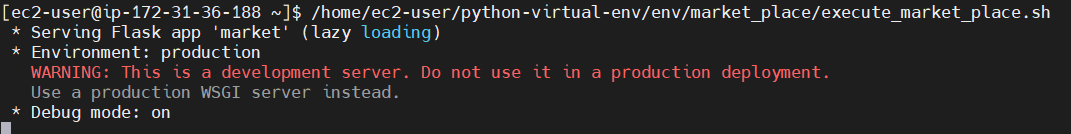
## How to Execute Python Flask File on AWS

1. Copy the public IP from the EC2 where the web application is located



1. Open new session to the copied ip using MobaExtrerm or other tool
2. Run the following **command /home/ec2-user/python-virtual-env/env/market\_place/execute\_market\_place.sh** which will start the web application

You shell see above output



1. Concatenate the port 8081 to the copied public IP , for instance <http://3.17.206.230:8081/>

In our case the copied ip is http://3.17.206.230/ and paste it in the browser:



