ClimateGuard: Paving the way to Eco-Friendly Living

Juyee Sabade, Dharmik Bhanushali, Shambhavi Shukla

Introduction

Objective

The primary objective of our project is to analyze daily carbon emissions originating from US domestic flights and the environmental impact associated with electricity, natural gas and coal consumption. Our goal is to promote sustainability by advocating for practices like reducing non-essential travel, opting for eco-friendly travel options, and selecting renewable and environmentally responsible services.

Motivation

Carbon emissions have consistently posed a significant threat to the environment, contributing to various issues like global warming, alterations in weather patterns, and rising average temperatures. Unfortunately, due to human negligence and a lack of sufficient attention to the matter, carbon emissions have become a pressing global concern. This issue urgently demands global attention. Through this project, our goal is to harness the capabilities of cloud computing services to analyze and mitigate carbon emissions associated with flights, and electricity and fossil fuels consumption. These factors collectively contribute to significant carbon footprints, and addressing them is crucial for environmental sustainability.

Scope

The scope of this project includes building an interactive dashboard that allows users to perform comprehensive data analysis and time series forecasting of carbon emissions related data, and study the long term adverse effects of carbon footprint on environment and ecology through a causal inference of the data. They can additionally educate themselves about the sustainability initiatives and green solutions for curbing these issues.

The following features will be incorporated in the dashboard:

1. Monitor and analyze carbon emissions associated with a broad range of emission generating activities in the modern world.
    - Aggregate the carbon footprint created by daily domestic flights running in the US through air traffic monitoring techniques and sophisticated methodologies to calculate greenhouse gas (CO_2, Nitrogen Oxide, water vapors, etc.) emissions by considering factors such as distance traveled, no of passengers, cargo load, etc.
    - Compute the real time aggregated carbon emissions based on the electricity consumption estimates of different US states.
2. Understand the correlation between fuel usage (Coal, Natural Gas, Petroleum) and CO_2 emissions associated with different sectors such as Commercial, Industrial, Residential, etc.

Stakeholders

The following diverse group of stakeholders can utilize our dashboard analytics to lead innovative sustainability initiatives in their respective sectors.

- **Aviation Sector**: Major airlines, regional airports, and industry associations play pivotal roles in emissions reduction within the domestic flight domain. Our flight emissions analytics can aid them in inspecting frequent flight routes and incorporating green solutions and optimization strategies that can help curb the massive carbon footprint generated by the aviation industry.
- **Environmental Advocates**: Environmental organizations and advocates champion sustainability, influencing both aviation and digital sectors. They can use our dashboard to influence sustainable and environmentally responsible initiatives on a government and organizational level, and direct funds to the right cause.
- **Regulatory Bodies**: Government agencies and regulators oversee compliance with environmental standards in aviation and internet governance. They can be persuaded to make strong regulatory laws that promote sustainability within these sectors.
- **Consumers and Users**: Individuals and organizations, as end-users, impact emissions through choices in travel and energy consumption activities, making them pivotal stakeholders in sustainability advocacy.

Dataset

We will be utilizing a wide range of datasets for the different components of our projects.

1. US Energy Information Administration (EIA) - We will be using the open sourced EIA data to calculate and visualize CO2 emissions and carbon coefficients by different fuels (Coal, Natural Gas, Petroleum) and sectors (Residential, Industrial, Commercial) in US states.
   https://www.eia.gov/opendata/browser/co2-emissions/co2-emissions-aggregates
2. US Environmental Protection Agency (EPA) - This dataset provides information about environmental activities that may affect air, water, and land anywhere in the United States. We will be mapping this information with carbon emissions and check if there is any correlation, and visualize the overall environmental impact.
   https://www.epa.gov/enviro/envirofacts-data-service-api
3. OpenSky Network API - This open source API helps retrieve live airspace information which in our case can help in monitoring daily flight traffic information domestically in the United States. This information will subsequently be used to calculate daily carbon emissions generated by flights in the US.
   https://openskynetwork.github.io/opensky-api/

Background and Related Work

Related Work

Below are some diverse studies and articles contributing and addressing carbon emissions while raising awareness for environmental sustainability. They highlight the importance of technology and innovation in making the world more environmentally responsible.

1. An emissions inventory using flight information reveals the long-term changes of aviation CO2 emissions in China

   https://www.sciencedirect.com/science/article/abs/pii/S0360544222023957

2. Reducing Carbon Emissions in websites:
   https://ieeexplore.ieee.org/document/10223952/authors#authors

3. Comparison between electric and hydrogen planes -
   https://ieeexplore.ieee.org/xpl/conhome/10048095/proceeding

4. Evaluation of Aviation Emissions and Environmental Costs in Europe Using OpenSky and OpenAP
   https://www.mdpi.com/2673-4591/13/1/5

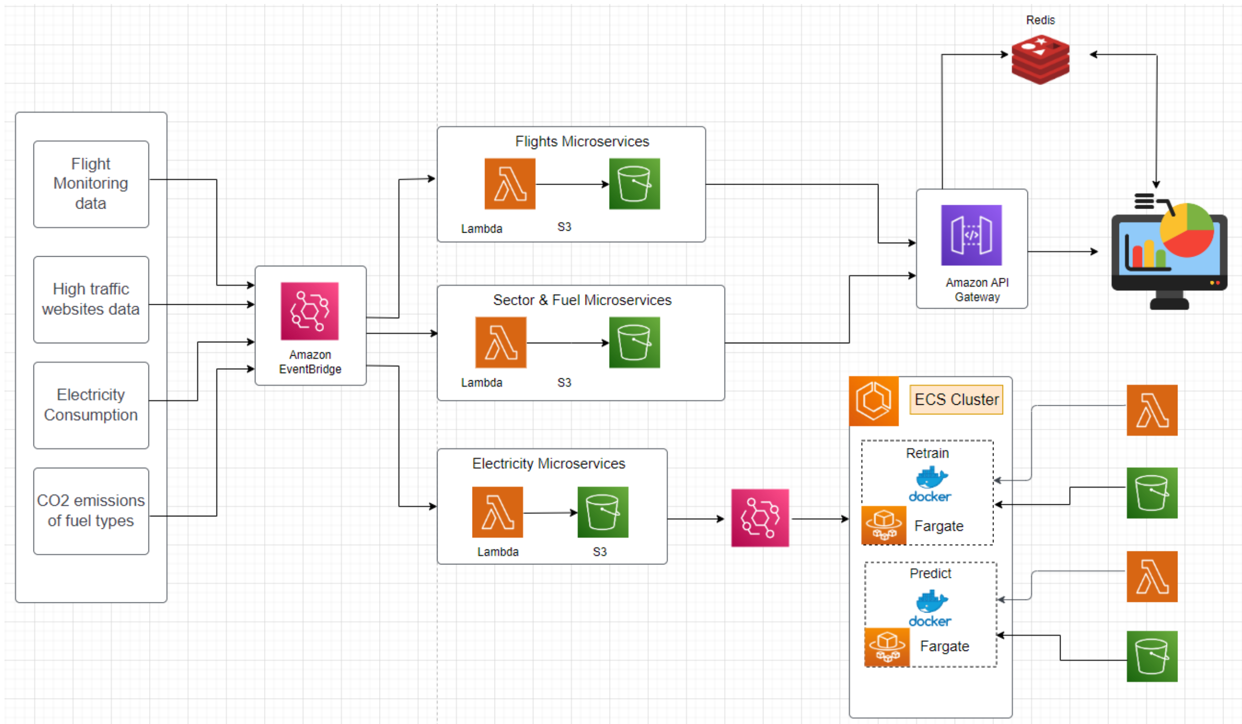
5. MIT - How to reduce your carbon footprints in virtual world:
   How to reduce the environmental impact of your next virtual meeting | MIT News | Massachusetts Institute of Technology

6. Carbon emission footprint on the internet -
   What is Your Online Carbon Footprint and How to Reduce it? | OVO Energy

Gap Analysis

At present, there is a lack of widespread awareness among individuals and organizations about the significant contributions of domestic flights, digital technologies, increasing coal and electricity to carbon emissions and climate change. There is also a lack of a centralized dashboard that documents and analyzes carbon emissions generated from different commercial/industrial/residential activities. Data related to it is dispersed making it difficult to analyze and evaluate. This project aims to bridge the existing gap by raising awareness and fostering a sense of responsibility among individuals and organizations, encouraging greener choices. The project seeks to create a cohesive and centralized platform for data integration, enabling comprehensive analysis of carbon emissions.

System Architecture



The above diagram depicts the system architecture for ClimateGuard. The components of this architecture include:

Data Collection

Data is collected from various third party REST API sources. We will be focusing on flight monitoring data, fuel & sector emissions data, and CO2 emissions generated from electricity consumption. For the purposes of this project, the region for all the data components will be restricted to the USA. The detailed methodologies and data source links to extract data are listed below:

1. **Air Traffic Monitoring & Carbon Emissions Calculation**
   In this task, we will be calculating aggregated carbon emissions generated by daily flight departures in frequently visited US airports. Due to API constraints and limited computational resources, we have shortlisted 30 domestic/international airports that have a high air traffic around the year and we will be focusing solely on US domestic departures in these airports.

   Data
   For airport traffic tracking the free, open source OpenSky Network API will be used.
   - Arrivals by Airport: https://openskynetwork.github.io/opensky-api/rest.html#arrivals-by-airport

- Departures by Airport:
  https://openskynetwork.github.io/opensky-api/rest.html#departures-by-airport

The following parameters are requested by the API:

- airport: ICAO identifier for the airport
- begin: Start of time interval to retrieve flights for as Unix time (seconds since epoch)
- end: End of time interval to retrieve flights for as Unix time (seconds since epoch)

The following table will be used to get the ICAO identifiers for all US airports.
https://github.com/ip2location/ip2location-iata-icao/blob/master/iata-icao.csv

The following websites provides detailed steps about calculating carbon emissions associated with a flight:
https://oncarbon.app/articles/how-you-calculate-flight-emissions

Process

- The API can fetch daily air traffic data with a delay of 1 day i.e the latest data that will be available today will be that of yesterday. Every time the dashboard is invoked, it will fetch the latest air traffic figures and calculate the carbon emissions generated from them.
- Since the data updates daily, the data is cached on the system first time the dashboard is run in the day, for latency optimization.
- We build an adjacency graph of flights for monitoring the current air traffic to make sure we are not double counting the carbon emissions.

2. Electricity Consumption Carbon Emissions Calculation

Data
https://www.eia.gov/opendata/browser/electricity/rto/daily-region-sub-ba-data

Process

- Monitor the electricity generated in US subregions on a daily basis.

3. CO2 Emissions Aggregate and C02 coefficients

Data
https://www.eia.gov/opendata/browser/co2-emissions/co2-emissions-aggregates
https://www.eia.gov/opendata/browser/co2-emissions/co2-emissions-and-carbon-coefficients
Data is available from 1970 to 2021.

Process

- Calculate CO2 emissions based on fuel, sector and US states.
  - Fuel type: Coal, Natural Gas, Petroleums
  - Sector: Commercial CO2 emissions, Industrial CO2 emissions, Transportation CO2 emissions, Electric power CO2 emissions, Residential CO2 emissions.

- Calculate electricity, coal, natural gas and petroleum consumption over the decades and map it with the C02 emissions being generated. Look at factors such as sulfur content, heat content etc. that harm the environment.
- Predict future CO2 emissions in different states based on current consumption patterns.

Data Transformation and Storage

The data from the API sources is extracted on lambda functions on a daily basis via Amazon EventBridge. Within these lambda functions, data cleaning, data manipulation and data transformation is done to get the desired results. Furthermore, these calculations are stored in Amazon S3 in a JSON format. The detailed methodologies for each task is detailed below:

Air traffic monitoring and carbon emission calculation

Step 1: Retrieve real time US domestic flights information

To do this, we're retrieving data from the Open Sky API that pertains to domestic flight departures in the US on a daily basis. This data includes information such as the time of the last aircraft sighting, the estimated arrival and departure airports, and the potential airport candidates around the source and destination airports for emergency landings in case such a situation arises.

The following API is being used to monitor domestic departures in US:

https://opensky-network.org/api/flights/departure?airport=[ICAO identifier of airport]&begin=[begin_timestamp]&end=[end_timestamp]

The following data will be used to get the ICAO identifiers for all US airports.

- https://github.com/ip2location/ip2location-iata-icao/blob/master/iata-icao.csv
- https://github.com/mwgg/Airports/blob/master/airports.json

We will be working with the top 50 busiest US airports with high to moderate traffic. The information is sourced from here:
https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_the_United_States

The following JSON object is an example of the result produced from the above API:


```
{
    "icao24": "ab6229",
```

"firstSeen": 1698288392,
"estDepartureAirport": "KORD",
"lastSeen": 1698293889,
"estArrivalAirport": "4OK1",
"callsign": "SWA3973 ",
"estDepartureAirportHorizDistance": 3334,
"estDepartureAirportVertDistance": 138,
"estArrivalAirportHorizDistance": 26396,
"estArrivalAirportVertDistance": 11329,
"departureAirportCandidatesCount": 521,
"arrivalAirportCandidatesCount": 0
}

Step 2: Calculate the distance between source and destination airport

The distance between source and destination airport derived from API is calculated using geographical coordinates of the airports and the distance obtained is in Kilometer (KM).The following code is used for the calculation.

```
def calculate_distance(x1,y1,x2,y2):
    R = 6373.0
    lat1 = radians(x1)
    lon1 = radians(y1)
    lat2 = radians(x2)
    lon2 = radians(y2)

    dlon = lon2-lon1
    dlat = lat2-lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2*atan2(sqrt(a),sqrt(1-a))

    return R*c
```

Step 3: Calculate the fuel consumption of each flight journey

From an aviation standpoint, fuel consumption is calculated using various factors such as fuel type, aircraft engine type, geographical conditions, aircraft dimensions and seating capacity, cruise speed, thrust, efficiency, etc.

However due to limitations of obtaining such aviation specific data, we will be assuming a standard aircraft type for all our flights to calculate the fuel consumption. We obtained information from the following source https://simpleflying.com/usa-domestic-widebody-flights/ regarding widely used aircrafts for domestic journeys in the US. It is stipulated that 23% of US airline companies use Boeing 767-300ER aircraft.

The specifications of this aircraft were obtained from https://en.wikipedia.org/wiki/Boeing_767#Specifications

Specifications of Boeing 767-300ER

Fuel Capacity	16,700–24,140 US gal
Thrust	252–274 kN
Cruise Speed	850-900 kmph
Engine Type	PW4000

To calculate fuel consumption of a flight journey, we will be using the concept of Thrust Specific Fuel Consumption, which is the amount of fuel consumed by an aircraft engine per unit time for a given thrust rating.

The formula for SFC is:

$$SFC = Fuel\ Burn \div Thrust$$

The SFC for engine PW4000 is 9.9-10.2g/kN s as obtained from https://en.wikipedia.org/wiki/Thrust-specific_fuel_consumption#Typical_values_of_SFC_for_thrust_engines.

The average thrust of the aircraft is 252-274kN. This is the thrust during cruise time as that is the maximum time of a commercial flight. This gives us fuel burn in unit g/s. It will be further converted to kg/hr.

Furthermore, cruise speed and distance is used to calculate the total duration of the flight.

$$Cruise\ Speed\ (kmph) = Distance\ (km) \div Time$$

This helps in calculating the total fuel consumption in unit Kg throughout the journey.

$$Total\ Fuel\ Consumption = Fuel\ consumed\ per\ hr * Time\ (hrs)$$

Step 4: Calculate the CO2 emissions generated by each flight journey

To calculate carbon emissions, we follow the methodology explained in the following source. https://oncarbon.app/articles/how-you-calculate-flight-emissions

- **Convert fuel burn to emissions**
  Burning one kilogram of jet fuel produces 3.157 kg of carbon dioxide. This ratio is used to calculate CO2 emissions associated with the flight journey.
- **Add emissions that result from fuel production**
  Fossil fuels do not just cause emissions when they are burned – large amounts of CO2 and other gasses are also released during extraction, drilling, refining, and transportation. To calculate that, we use a constant of 0.617 kg of CO2 for these emissions per kilogram of jet fuel produced.

The following code will be used to calculate the fuel consumption and carbon emissions of all flight journeys in the unit kg/hr.

```
def calculate_emissions(dist):
    sfc = 9.9
    thrust = 252

    #Thrust Specific fuel consumptions (TSFC) is fuel consumption (grams/second) per unit of
thrust (newtons, or N)
    fuel_burn = sfc * thrust
    #Converting fuel burn from g/s to kg/hr
    fuel_burn_kg_hr = (fuel_burn/1000)*3600

    #Calculating time taken for the journey based on airplane's average cruise speed in km/hr
    cruise_speed = 900
    time = dist/cruise_speed

    total_fuel_burn = fuel_burn_kg_hr * time
    #Burning one kilogram of jet fuel produces 3.157 kg of carbon dioxide
    co2_emissions = total_fuel_burn * 3.157
    #Calculating the amount of CO2 emissions generated during fuel production
    fuel_production_emissions = total_fuel_burn * 0.617
    total_co2_emissions = co2_emissions + fuel_production_emissions

    return total_co2_emissions
```

Electricity Consumption

In our second task, we are consistently gathering data on the electricity consumption of various states in the United States. This information is updated monthly through AWS Lambda functions. By utilizing this fresh data, we can calculate the carbon emissions specific to each state and the corresponding time period.

Step 1: Fetch the data using the below API

[https://api.eia.gov/v2/electricity/electric-power-operational-data/data/?frequency=monthly&data[0]=consumption-for-eg&data[1]=consumption-for-eg-btu&data[2]=consumption-uto-btu&data[3]=sulfur-content&facets[location]](https://api.eia.gov/v2/electricity/electric-power-operational-data/data/?frequency=monthly&data[0]=consumption-for-eg&data[1]=consumption-for-eg-btu&data[2]=consumption-uto-btu&data[3]=sulfur-content&facets[location])

Sample API response:


```
{
    "period": "2023-12-06",
    "subba": "ZONG",
    "subba-name": "Hudson Valley",
    "parent": "NYIS",
    "parent-name": "New York Independent System Operator",
    "timezone": "Eastern",
    "value": 27508,
    "value-units": "megawatthours"
}
```


Step 2: Implement this API using AWS lambda to retrieve data in regular intervals and keep updating the CO2 emissions.

Step 3: Calculate the carbon emissions

Carbon Emissions (in metric tons of CO2) = Electricity Consumption (in megawatt-hours, MWh) × Carbon Intensity (in kilograms of CO2 per MWh)

1. Electricity Consumption (MWh): This is the total amount of electricity consumed by a specific sector over a certain period. You can obtain this information from your electricity bills or the relevant government or industry statistics.

2. Carbon Intensity (kg of CO2 per MWh): This is a measure of how much carbon dioxide (CO2) is emitted per unit of electricity generated. It can vary significantly depending on the energy sources used to generate the electricity. You can typically find this information from your electricity provider, government agencies, or environmental organizations.

CO2 aggregate emission by fuel and sector in US states

For our third task, we will be obtaining data on carbon dioxide (CO2) emissions resulting from different fuels usage in various sectors. This data will be collected on a monthly basis. It will provide us with the total CO2 emissions, the specific time period during which these emissions occurred, and the sectors responsible for these emissions, along with the corresponding CO2 emission quantities.

Step 1: Fetch the data from below API

API:
[https://api.eia.gov/v2/co2-emissions/co2-emissions-aggregates/data/?frequency=annual&data[0]=value&sort[0][column]=period&sort[0][direction]=desc&offset=0&length=5000](https://api.eia.gov/v2/co2-emissions/co2-emissions-aggregates/data/?frequency=annual&data[0]=value&sort[0][column]=period&sort[0][direction]=desc&offset=0&length=5000)

Sample API response:


```json
{
    "period": 2021,
    "sectorId": "RC",
    "sector-name": "Residential carbon dioxide emissions",
    "fuelId": "PE",
    "fuel-name": "Petroleum",
    "stateId": "AL",
    "state-name": "Alabama",
    "value": 0.295148,
    "value-units": "million metric tons of CO2"
 }
```


The parameters of the API include

- Fuel type - Coal, Petroleum, Natural gas
- Sector Type - Commercial, Industrial, Residential, Transportation
- US State

Step 2: Implement the lambda function to fetch data from this API in regular intervals to keep updating the CO2 emissions.

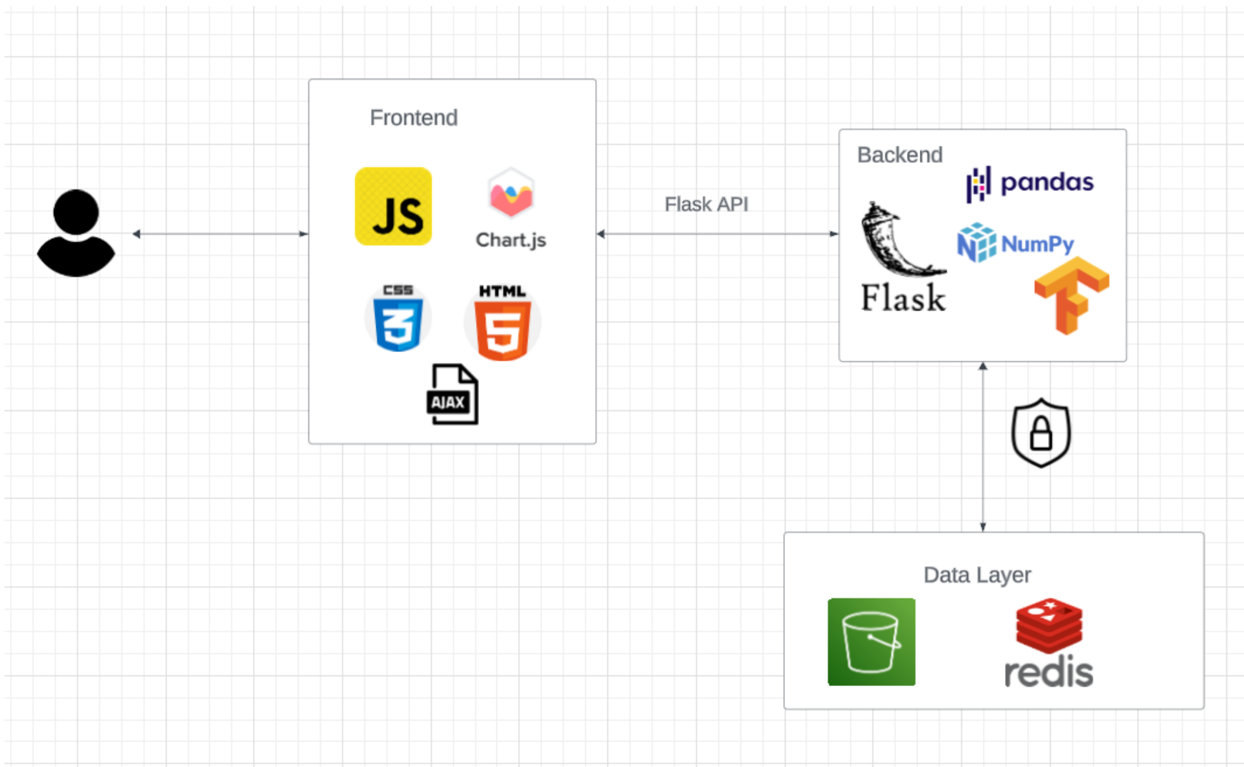
```
headers = {
            "X-Params": json.dumps({
               "frequency": "annual",
               "data": ["value"],
               "facets": {
```

```
            "sectorId": [
                "CC",
                "IC",
                "RC",
                "TT"
            ]
        },
        "start": 1970,
        "end": 2021,
        "sort": [{"column": "period", "direction": "desc"}],
        "length": 5000,
        "offset":0
    }),
    "User-Agent": "Your User Agent Here",
    "Content-Type": "application/json"
}

while start_flag or len(emissions_data['response']['data'])>0:
    start_flag=False
    response = requests.get(api_url, params = params, headers=headers)
    if response.status_code == 200:
        emissions_data = response.json()
        #print(emissions_data['response']['total'])
        for doc in emissions_data['response']['data']:
            fuel_emissions.append(doc)
        counter+=5000
        headers = {
            "X-Params": json.dumps({
                "frequency": "annual",
                "data": ["value"],
                "facets": {
                    "sectorId": [
                        "CC",
                        "IC",
                        "RC",
                        "TT"
                    ]
                },
                "start": 1970,
                "end": 2021,
                "sort": [{"column": "period", "direction": "desc"}],
                "length": 5000,
                "offset":counter
            }),
            "User-Agent": "Your User Agent Here",
            "Content-Type": "application/json"
        }
    else:
        print("Failed to retrieve data. Status code:",response.status_code)
```

Data Analysis & Visualization

For analyzing the data, we have built an interactive dashboard with a flask backend. The architecture of our dashboard is shown below.



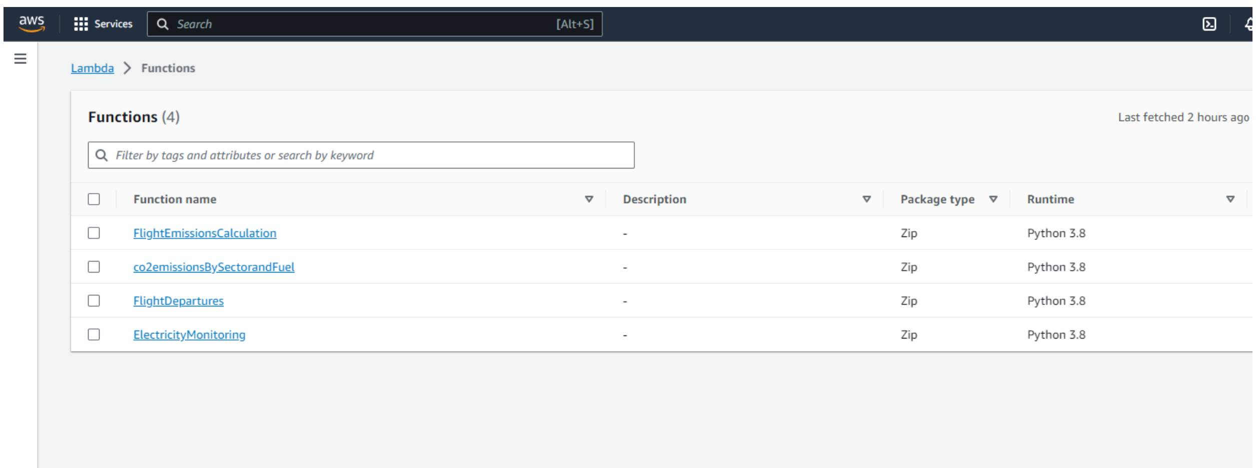
Redis Caching

We have implemented Redis for caching the result calculations during daily runs, so that if the dashboard is run multiple times a day, the result will be extracted from the Redis Cache.

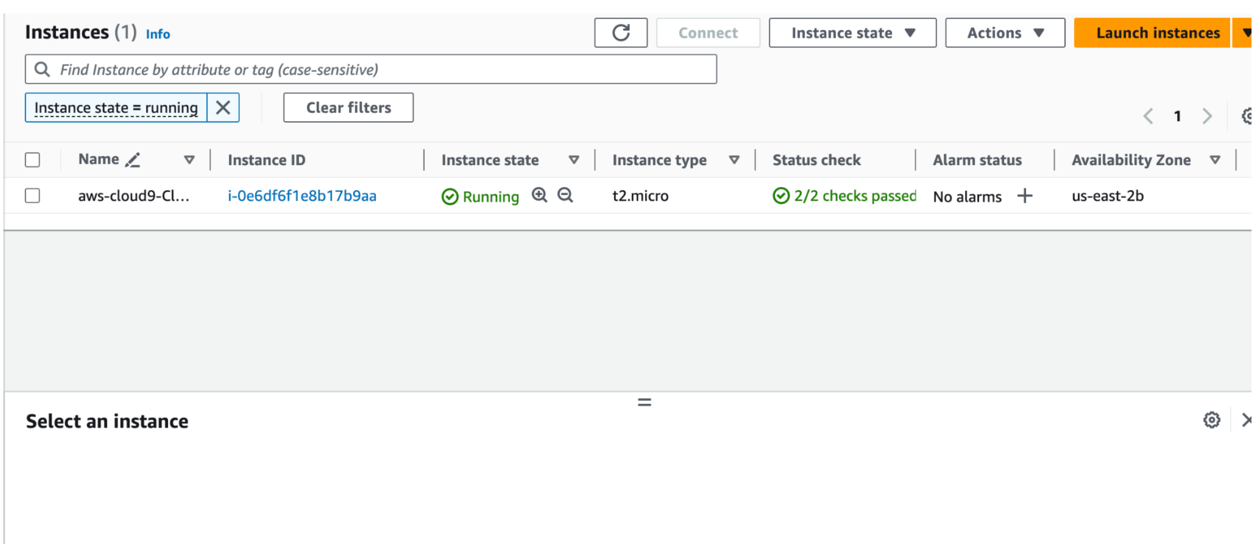
Results

Lambda Function Results

We are currently receiving data from multiple API functions and calls, each of which corresponds to one of our proposed tasks. To automate this data retrieval, we have set up Lambda functions that will be triggered at regular intervals, depending on our scheduling decisions, which could be daily, weekly, or monthly.



To support the lambda functions, we've established an EC2 instance that enables us to install and manage the necessary libraries and dependencies in the cloud. These are integrated into the lambda functions as lambda layers.

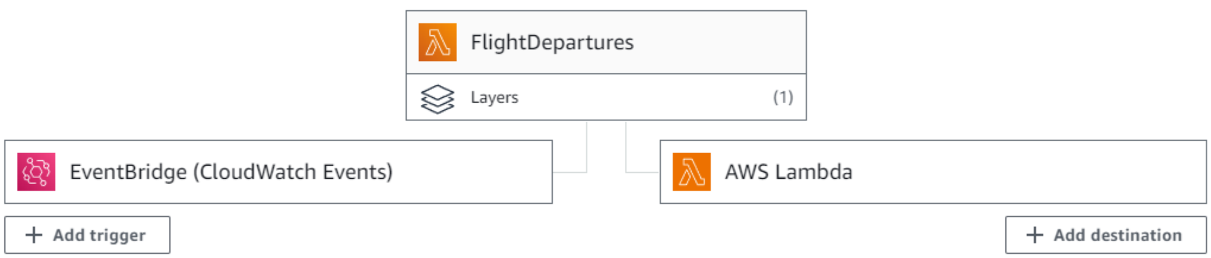


As part of our data management strategy, we are in the process of storing this data in an S3 bucket. This approach not only facilitates data storage and management but also allows us to implement caching mechanisms to minimize unnecessary API calls. We are storing the data from our APIs in S3 buckets. Storing API data in S3 buckets in JSON format benefits our project in multiple ways. It ensures scalability for growing data needs, data durability, efficient management, high availability, cost-effectiveness, and data security. The caching option reduces API calls, improving performance and lowering costs. S3's easy integration with cloud services further enhances our project's functionality.

The following lambda functions have been created:

1. Flight Departure Monitoring

Architecture



Objective

The objective of this function is to monitor domestic flight departures in the US via API. It's triggered to run on a daily basis at 12 AM UTC via Amazon EventBridge. The delay of the API information is 1 day and the window is 1 day i.e latest flight departure information from day after to yesterday is obtained.

Input Data

The following API is being used to monitor domestic departures in US:

[https://opensky-network.org/api/flights/departure?airport=[ICAO identifier of airport]&begin=[begin_timestamp]&end=[end_timestamp]](https://opensky-network.org/api/flights/departure?airport=[ICAO identifier of airport]&begin=[begin_timestamp]&end=[end_timestamp])

The following data will be used to get the ICAO identifiers for all US airports.

- https://github.com/ip2location/ip2location-iata-icao/blob/master/iata-icao.csv
- https://github.com/mwgg/Airports/blob/master/airports.json

Result

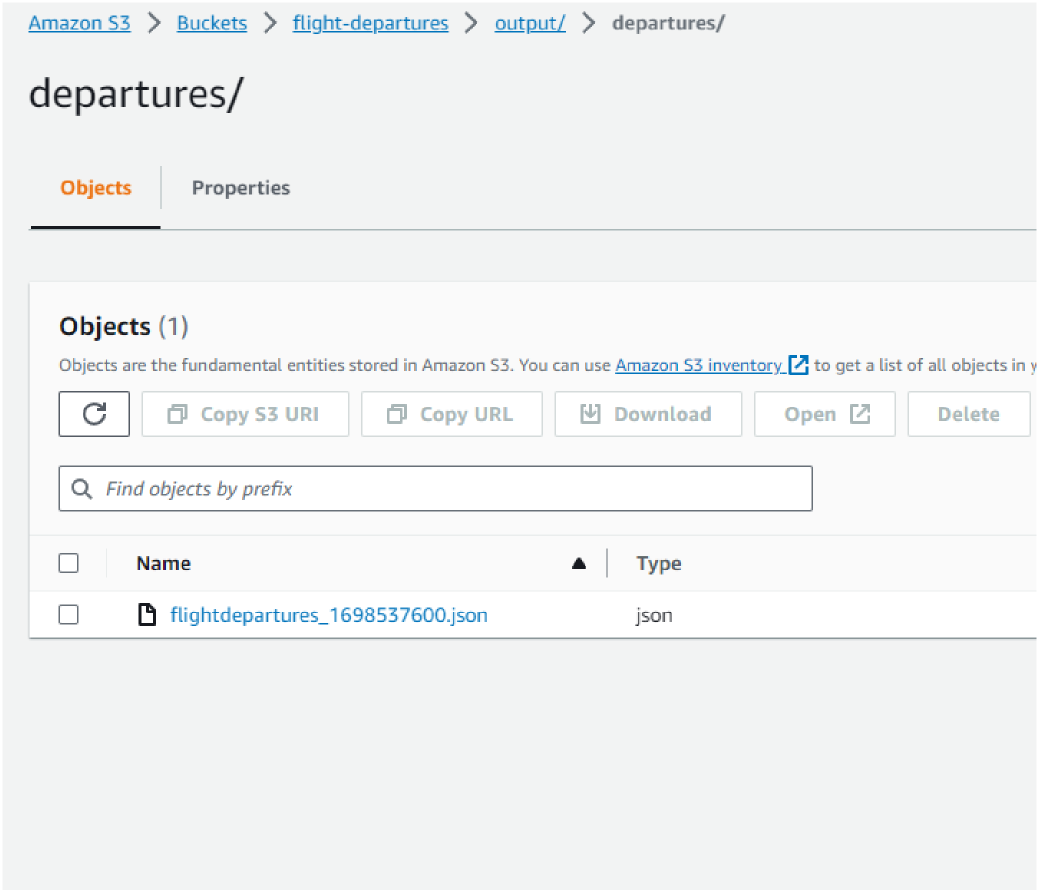
The result is obtained in a JSON format and is stored in a S3 bucket.

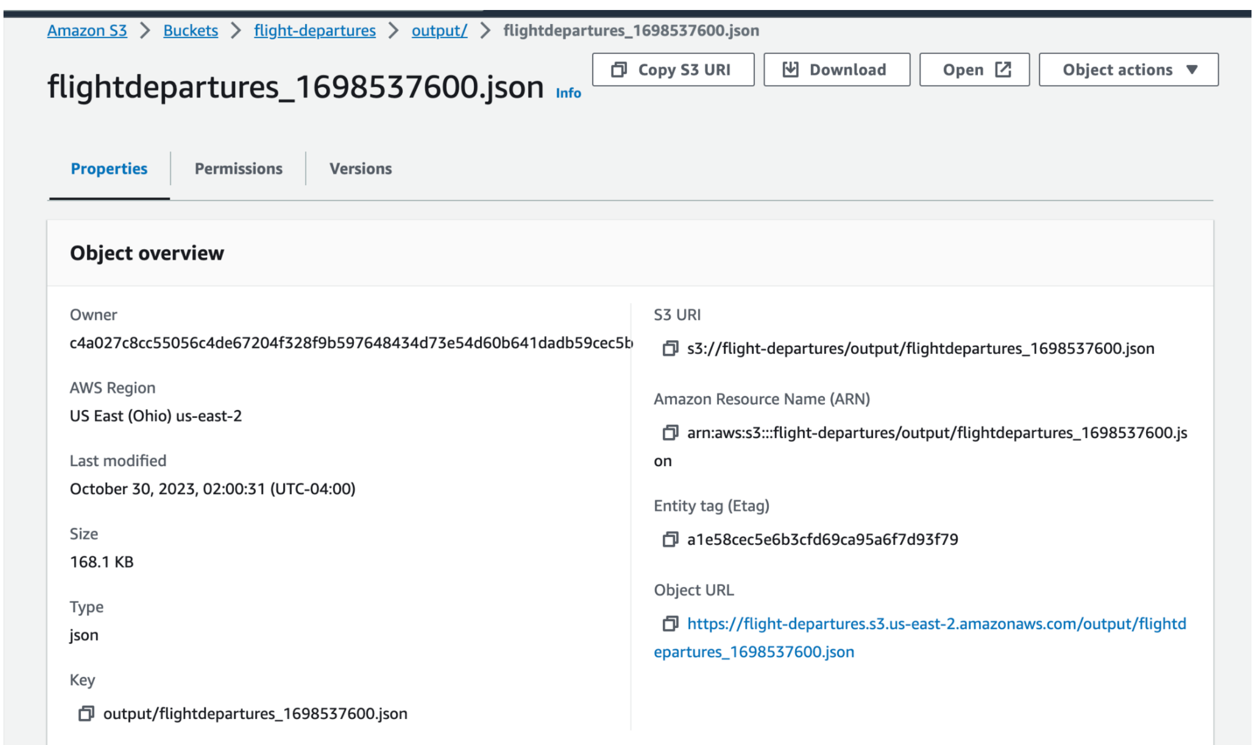
Result Format:


```
{
    data: [
        {
            "icao24": "ab6229",
            "firstSeen": 1698288392,
            "estDepartureAirport": "KORD",
            "lastSeen": 1698293889,
            "estArrivalAirport": "4OK1",
            "callsign": "SWA3973 ",
            "estDepartureAirportHorizDistance": 3334,
            "estDepartureAirportVertDistance": 138,
            "estArrivalAirportHorizDistance": 26396,
            "estArrivalAirportVertDistance": 11329,
            "departureAirportCandidatesCount": 521,
            "arrivalAirportCandidatesCount": 0
        }
    ],

    count: 1 //to document the count of records in the given timeframe
}
```


Following is a snippet of the result stored in a S3 bucket ‘flight-departures’.


Amazon S3 > Buckets > flight-departures > output/ > departures/
departures/
Objects
Properties
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in y
Copy S3 URI
Copy URL
Download
Open
Delete
Find objects by prefix
Name
Type
flightdepartures_1698537600.json
json


Amazon S3 > Buckets > flight-departures > output/ > flightdepartures_1698537600.json
flightdepartures_1698537600.json
Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
c4a027c8cc55056c4de67204f328f9b597648434d73e54d60b641dadb59cec5b
AWS Region
US East (Ohio) us-east-2
Last modified
October 30, 2023, 02:00:31 (UTC-04:00)
Size
168.1 KB
Type
json
Key
output/flightdepartures_1698537600.json
S3 URI
s3://flight-departures/output/flightdepartures_1698537600.json
Amazon Resource Name (ARN)
arn:aws:s3:::flight-departures/output/flightdepartures_1698537600.json
Entity tag (Etag)
a1e58cec5e6b3cfd69ca95a6f7d93f79
Object URL
https://flight-departures.s3.us-east-2.amazonaws.com/output/flightdepartures_1698537600.json

2. Flight Emissions Calculation

Objective

The objective of this function is to calculate the carbon emissions associated with the flight journeys in the US that we obtained from the previous function. It is triggered to run on a daily basis after the successful completion of the flight departures monitoring lambda function. This data can help us in visualizing the carbon emissions generated in the atmosphere in the US from just domestic flights. The methodology to calculate carbon emissions has been described above.

Input Data

The following data will be used to get the ICAO identifiers, airport information and geographical coordinates for all US airports.

- https://github.com/ip2location/ip2location-iata-icao/blob/master/iata-icao.csv
- https://github.com/mwgg/Airports/blob/master/airports.json

Furthermore, the JSON output of the previous lambda function is the input to this function.

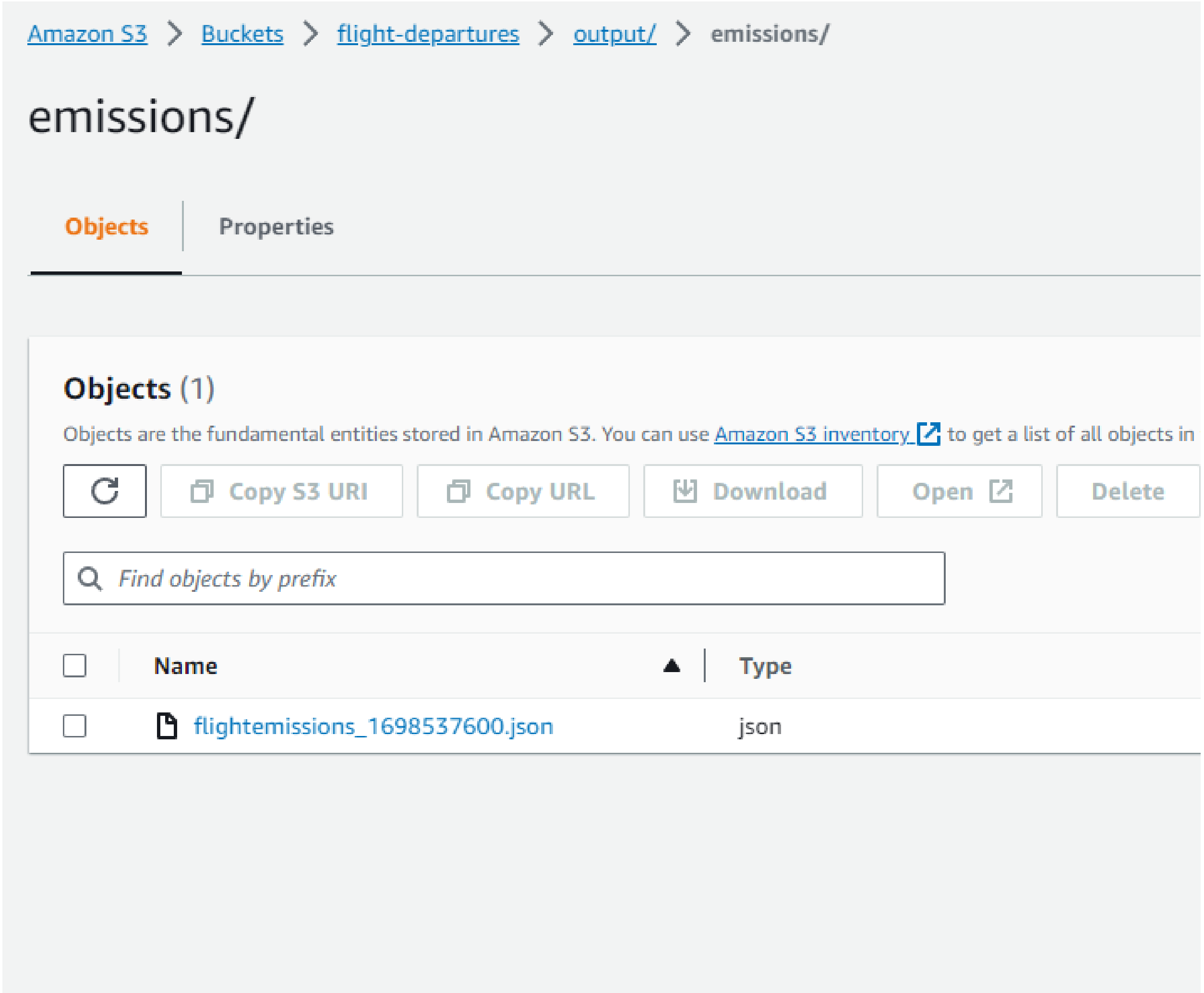
Result

The output is a list of JSON objects with each object documenting the carbon emissions of the particular flight journey.

Here's an example of the output format:

```
{
    departureAirportName: “Salt Lake City International Airport”,
    departureAirportCountry: “US”,
    arrivalAirportName: “Norman Y. Mineta San Jose International Airport”,
    arrivalAirportCountry: “US”,
    distanceBetweenAirport: 939.22,
    distanceUnit: “KM”,
    CO2Emissions: 35372,
    emissionsUnit: “KG”
}
```

Following is a snippet of the result stored in a S3 bucket ‘flight-departures’.

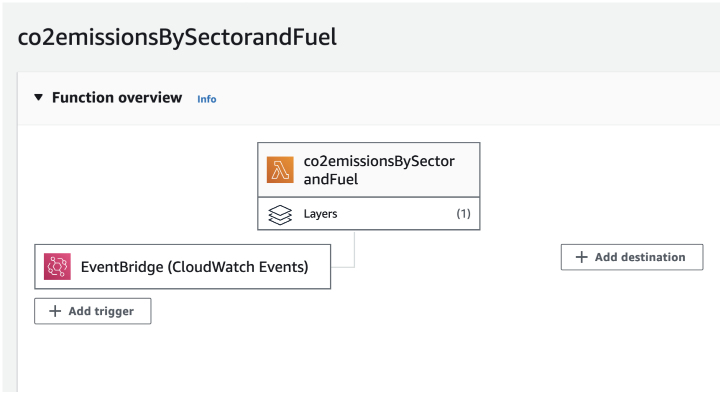


3. CO2 aggregate emissions by sector and fuel

Objective:

The objective of this function is to collect the CO2 emissions from different sectors and the fuel responsible for that emission for multiple states and regions in the US and analyze how that data and CO2 emissions has been varying over the decades.

Architecture:



Result

The output is a list of JSON objects with each object documenting the carbon emissions of the fuel & sector in a particular state in a particular year.

Here's an example of the output format:

{
'period': 2021,
'sectorId': 'RC',
'sector-name': 'Residential carbon dioxide emissions',
'fuelId': 'CO',
'fuel-name': 'Coal',
'stateId': 'SC',
'state-name': 'South Carolina',
'value': 0.024819,
'value-units': 'million metric tons of CO2'
}

Codebase

https://github.com/juyee1698/ClimateGuard

Dashboard Demo

We have implemented CO2 emissions from US domestic flights and sector and fuel usage.

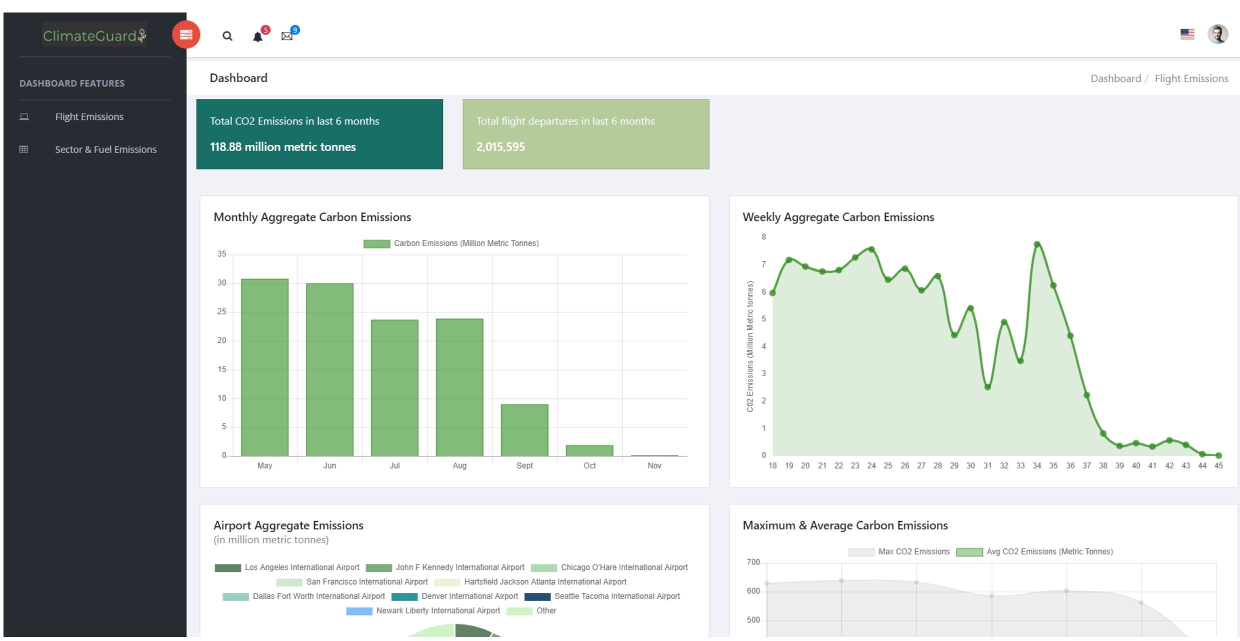
Dashboard Demo:
https://drive.google.com/file/d/1psfVZapNOmH2FOUgYPg2b6H-GaIpQd1F/view?usp=sharing

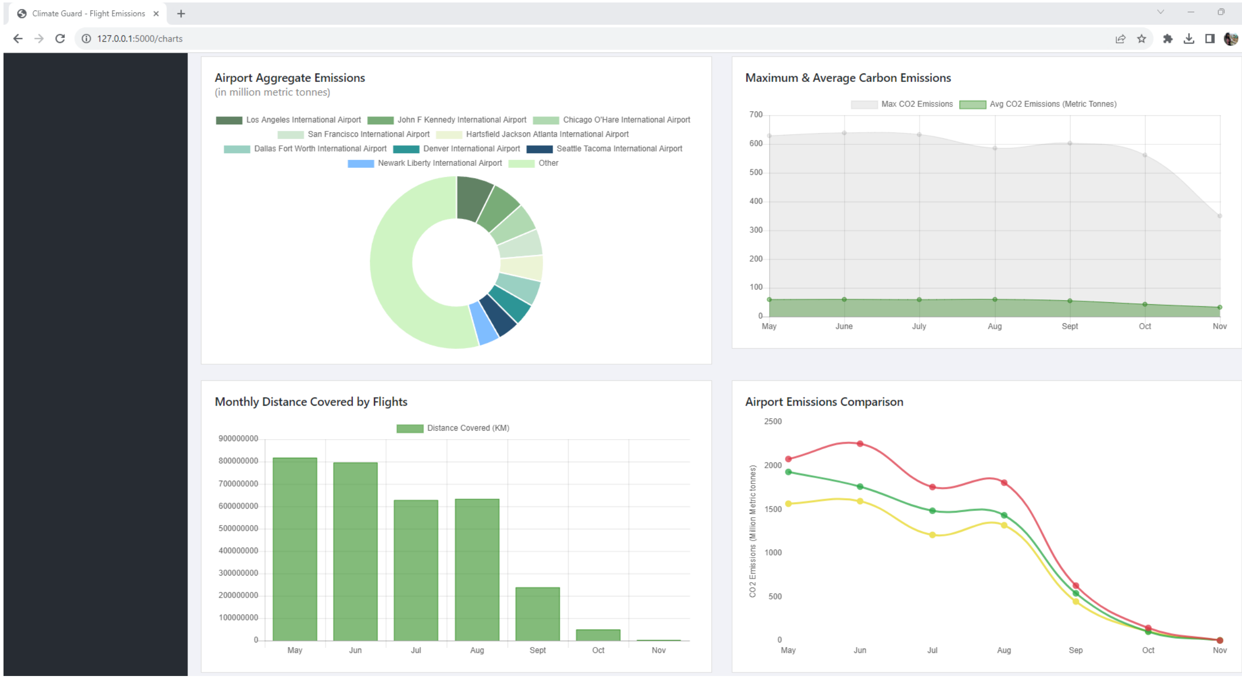
Snippets:

Flight Emissions

The following features have been added:

- Total CO2 emissions generated in the last 6 months
- Total flights departed in last 6 months
- Monthly Aggregate Emissions from US flight departures
- Weekly Aggregate Emissions from US flight departures
- Maximum and Average Carbon Emissions on a monthly basis
- Airport Aggregate Emissions of top 10 airports with largest flight traffic
- Comparison of top 3 airports with largest amount of flight traffic and emissions generation on a monthly basis.
- Monthly distance covered by flights.
- Airport monthly statistics (total emissions) by month of top 10 airports with largest flight traffic and emissions generation.





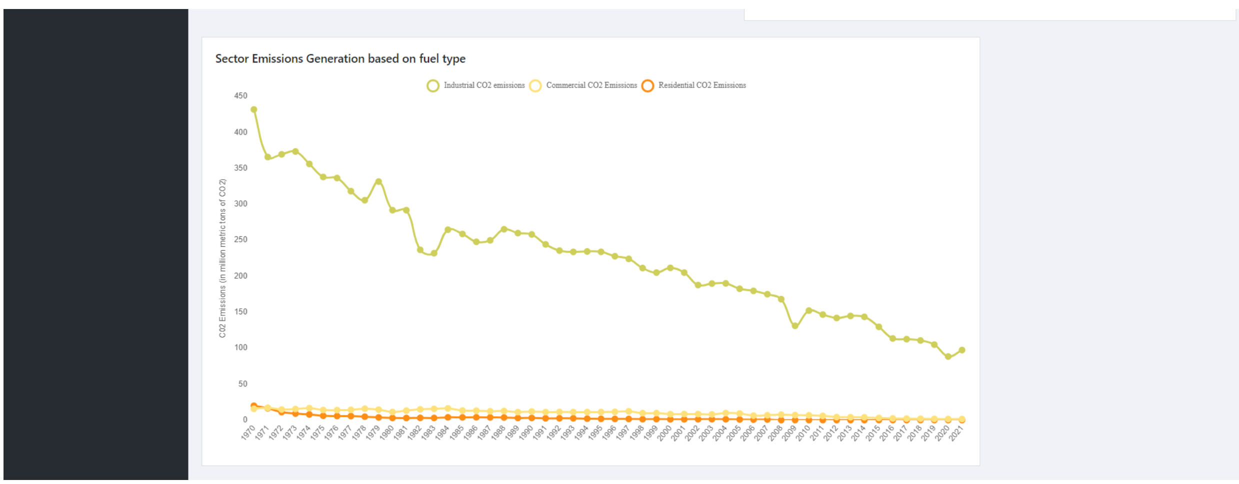
Airport Monthly Statistics

Departure Airport	May	Jun	Jul	Aug	Sep	Oct	Nov
Chicago O'Hare International Airport	1,566,861.9 tons	1,596,010.7 tons	1,210,021.7 tons	1,320,202.7 tons	447,064.3 tons	105,754.6 tons	3,391.1 tons
Dallas Fort Worth International Airport	1,406,404.7 tons	1,434,895.7 tons	1,155,258.9 tons	1,103,631.5 tons	431,652.9 tons	128,579.6 tons	5,090.1 tons
Hartsfield Jackson Atlanta International Airport	1,506,124.2 tons	1,475,373.3 tons	1,186,639.5 tons	1,143,730.8 tons	446,221.7 tons	81,877.6 tons	1,692.5 tons
John F Kennedy International Airport	1,931,020.2 tons	1,762,763.5 tons	1,486,153.8 tons	1,434,506.9 tons	541,358.3 tons	101,396.0 tons	2,678.0 tons
Los Angeles International Airport	2,078,484.8 tons	2,253,552.2 tons	1,758,171.5 tons	1,807,552.7 tons	628,947.7 tons	144,545.9 tons	1,072.1 tons
San Francisco International Airport	1,551,651.6 tons	1,531,106.8 tons	1,174,253.8 tons	1,195,710.8 tons	377,417.7 tons	82,561.5 tons	1,684.4 tons
Seattle Tacoma International Airport	1,228,393.0 tons	1,252,515.5 tons	1,055,863.3 tons	992,577.4 tons	375,531.1 tons	82,059.9 tons	4,175.8 tons

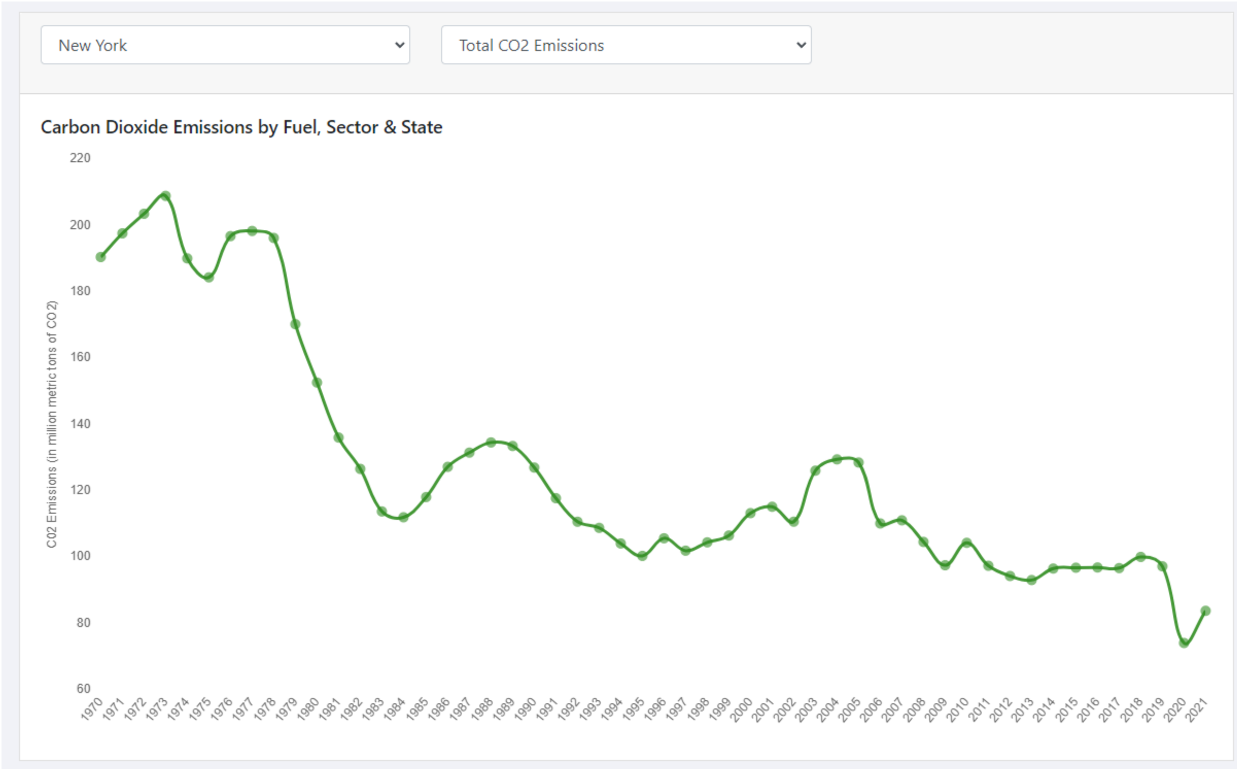
Sector & Fuel Emissions

The following features have been added:

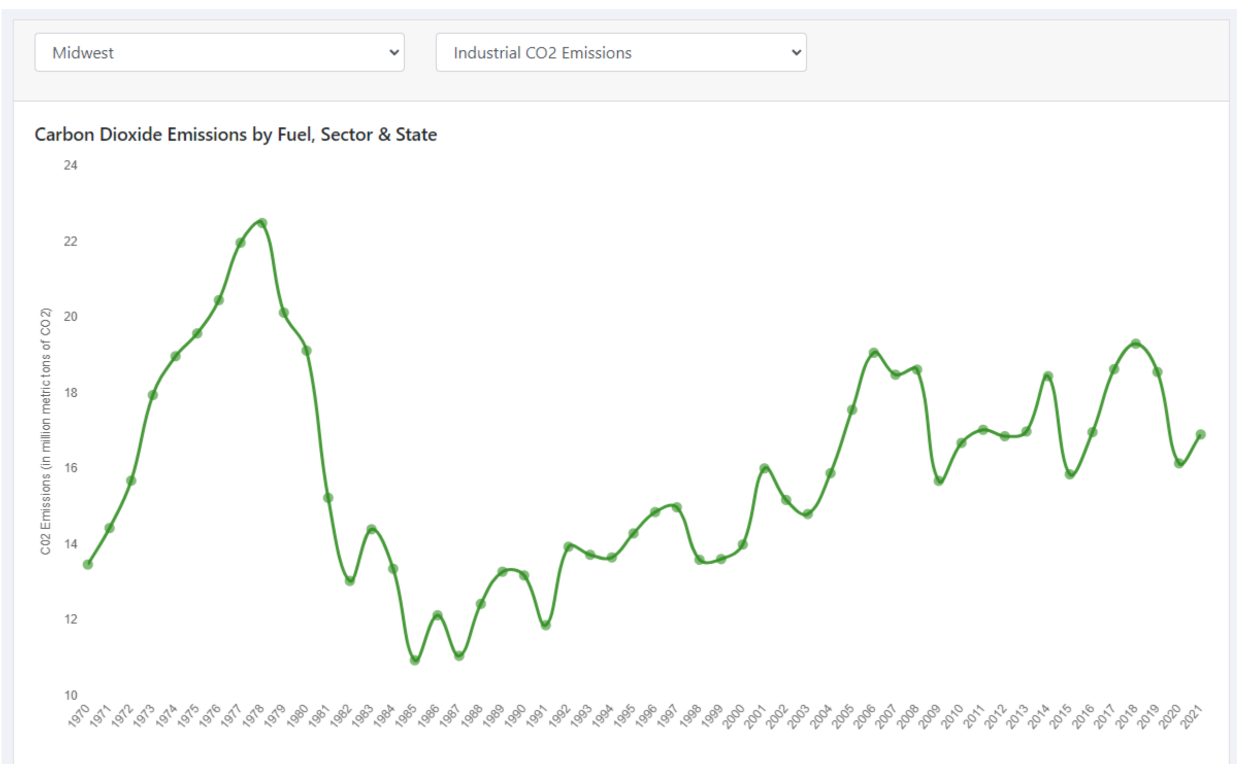
- The dashboard is interactive based on the fuel type chosen - All fuels, Coal, Petroleum, Natural Gas.
- CO2 Emissions calculation based on fuel, sector and US state.
- US Region statistics of carbon emissions based on fuel type chosen.
- Sector statistics (All sectors, Residential, Industrial,Commercial) based on fuel type chosen.
- Sector emissions generation on a yearly basis.
- Individual insights of carbon emissions generated by different fuel types from 1970 to 2021.
- Comparison of fuel emissions in the last 10 years.



Visualizing the first graph by state and sector.



New York
Total CO2 Emissions
Carbon Dioxide Emissions by Fuel, Sector & State
C02 Emissions (in million metric tons of CO2)
220
200
180
160
140
120
100
80
60
1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021



Midwest
Industrial CO2 Emissions
Carbon Dioxide Emissions by Fuel, Sector & State
C02 Emissions (in million metric tons of CO2)
24
22
20
18
16
14
12
10
1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021

Future Work

As part of future work, we would like to incorporate the following characteristics:

- Build a comprehensive and interactive dashboard which provides a centralized platform to visualize carbon emissions from various sectors such as flights, fuel, websites, cloud platforms, electricity generation, vehicles, etc.
- Build a real time streaming data pipeline that automatically extracts latest data from all our sources on an hourly/daily basis.
- Incorporate predictive analytics and time series forecasting models in our dashboard.
- Calculate the impact of CO2 emissions on the environment by assessing the meteorological patterns of different US locations such as heatwaves, floods and glacier melts, and ecological imbalances.
- Utilize our dashboard to influence sustainable and environmentally responsible initiatives on a government and organizational level, and direct funds to the right cause.
- Educate the public on the ongoing climate change crisis through the massive amounts of carbon emissions being generated daily, and influence them to take sustainable actions on an individual/community level by collaborating with non-profit environmental organizations.
- Integrate green solutions and alternatives on common carbon emission generating activities and provide social benefits to citizens who incorporate these green choices.