Deliverable 3. Create a Travel Itinerary Map.

In [1]:
```
# Dependencies and Setup
import pandas as pd
import requests
import gmaps

# Import API key
import sys
sys.path.append("../")
from config import g_key

# Configure gmaps
gmaps.configure(api_key=g_key)
```

In [2]:
```
# 1. Read the WeatherPy_vacation.csv into a DataFrame.
vacation_df = pd.read_csv("../Vacation_Search/WeatherPy_vacation.csv")
vacation_df.head()
```

Out[2]:

	City_ID	City	Country	Max Temp	Current Description	Lat	Lng	Hotel Name
0	0	Kalmunai	LK	77.18	overcast clouds	7.4167	81.8167	VS Villa
1	5	Hilo	US	71.60	overcast clouds	19.7297	-155.0900	Hilo Hawaiian Hotel
2	7	Sampit	ID	74.10	moderate rain	-2.5333	112.9500	Aquarius Boutique Hotel Sampit
3	8	Victoria	HK	75.99	few clouds	22.2855	114.1577	Mini Hotel Central
4	9	Carnarvon	AU	71.60	scattered clouds	-24.8667	113.6333	Hospitality Carnarvon

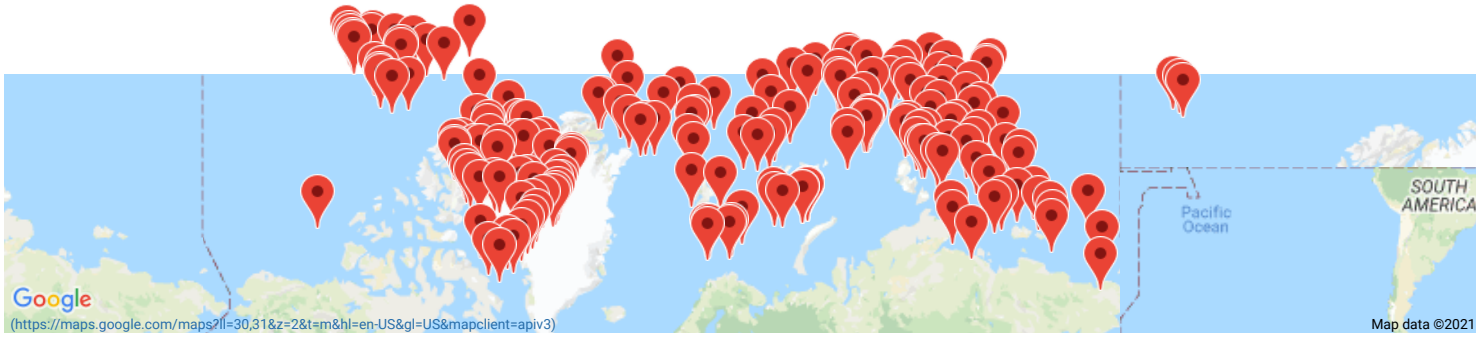
In [3]:
```
vacation_df.isnull().sum()
vacation_df.count()
clean_hotel_df = vacation_df.dropna()
#clean_hotel_df
```

In [4]:
```
# 2. Using the template add the city name, the country code, the weather description and maximum temperature for the city.
info_box_template = """
<dl>
<dt>City</dt><dd>{City}, {Country}</dd>
<dt>Weather</dt><dd>{Current Description}, {Max Temp} °F</dd>
</dl>
"""

# 3a. Get the data from each row and add it to the formatting template and store the data in a list.
hotel_info = [info_box_template.format(**row) for index, row in clean_hotel_df.iterrows()]

# 3b. Get the latitude and longitude from each row and store in a new DataFrame.
locations = clean_hotel_df[["Lat", "Lng"]]
```

In [5]:
```
# 4a. Add a marker layer for each city to the map.
locations = clean_hotel_df[["Lat", "Lng"]]
max_temp = clean_hotel_df["Max Temp"]
fig = gmaps.figure(center=(30.0, 31.0), zoom_level=1.5)
#heat_layer = gmaps.heatmap_layer(locations, weights=max_temp,dissipating=False,
#             max_intensity=300, point_radius=4)
marker_layer = gmaps.marker_layer(locations, info_box_content=hotel_info)
#fig.add_layer(heat_layer)
fig.add_layer(marker_layer)
# 4b. Display the figure
fig
```



In [ ]:
```
1
```

In [6]:
```
# From the map above pick 4 cities and create a vacation itinerary route to travel between the four cities.
# 5. Create DataFrames for each city by filtering the 'vacation_df' using the loc method.
# Hint: The starting and ending city should be the same city.

vacation_start = vacation_df.loc[vacation_df["City"] == "Kununurra"]
vacation_end   = vacation_df.loc[vacation_df["City"] == "Kununurra"]

vacation_stop1 = vacation_df.loc[vacation_df["City"] == "Mount Isa"]
vacation_stop2 = vacation_df.loc[vacation_df["City"] == "Emerald"]
vacation_stop3 = vacation_df.loc[vacation_df["City"] == "Yulara"]
```

In [7]:
```
# 6. Get the latitude-longitude pairs as tuples from each city DataFrame using the to_numpy function and list indexing.
start = vacation_start["Lat"].to_numpy()[0], vacation_start["Lng"].to_numpy()[0]
end =   vacation_end["Lat"].to_numpy()[0], vacation_end["Lng"].to_numpy()[0]
stop1 = vacation_stop1["Lat"].to_numpy()[0], vacation_stop1["Lng"].to_numpy()[0]
stop2 = vacation_stop2["Lat"].to_numpy()[0], vacation_stop2["Lng"].to_numpy()[0]
stop3 = vacation_stop3["Lat"].to_numpy()[0], vacation_stop3["Lng"].to_numpy()[0]
```

In [8]:
```
stop3
```

Out[8]: (-25.2406, 130.9889)

In [9]:
```
# 7. Create a direction layer map using the start and end latitude-longitude pairs,
# and stop1, stop2, and stop3 as the waypoints. The travel_mode should be "DRIVING", "BICYCLING", or "WALKING".
fig = gmaps.figure(center=(-20.0, 135.0), zoom_level=5.5)
city_itinerary = gmaps.directions_layer(
        start, end, waypoints=[stop1, stop2, stop3],
        travel_mode='DRIVING')

fig.add_layer(city_itinerary)
fig

```

In [10]:
```
# 8. To create a marker layer map between the four cities.
#  Combine the four city DataFrames into one DataFrame using the concat() function.
itinerary_df = pd.concat([vacation_start, vacation_stop1, vacation_stop2, vacation_stop3],ignore_index=True)
itinerary_df
```

Out[10]:

	City_ID	City	Country	Max Temp	Current Description	Lat	Lng	Hotel Name
0	249	Kununurra	AU	78.8	overcast clouds	-15.7667	128.7333	Hotel Kununurra
1	440	Mount Isa	AU	77.0	clear sky	-20.7333	139.5000	ibis Styles Mt Isa Verona
2	526	Emerald	AU	77.0	few clouds	-23.5333	148.1667	The Irish Village
3	324	Yulara	AU	71.6	clear sky	-25.2406	130.9889	Desert Gardens Hotel - Ayers Rock Resort

In [11]:

```
# 9 Using the template add city name, the country code, the weather description and maximum temperature for the city.
info_box_template = """
<dl>
<dt>Hotel Name</dt><dd>{Hotel Name}</dd>
<dt>City</dt><dd>{City}</dd>
<dt>Country</dt><dd>{Country}</dd>
<dt>Current Weather</dt><dd>{Current Description} and {Max Temp} °F</dd>
</dl>
"""

# 10a Get the data from each row and add it to the formatting template and store the data in a list.
hotel_info = [info_box_template.format(**row) for index, row in itinerary_df.iterrows()]

# 10b. Get the latitude and longitude from each row and store in a new DataFrame.
locations = itinerary_df[["Lat", "Lng"]]
locations #The latitude and longitude pairs for each of the four cities are retrieved. (5 pt)
```

Out[11]:

	Lat	Lng
0	-15.7667	128.7333
1	-20.7333	139.5000
2	-23.5333	148.1667
3	-25.2406	130.9889

In [12]:

```
# 11a. Add a marker layer for each city to the map.
locations = itinerary_df[["Lat", "Lng"]]
max_temp = itinerary_df["Max Temp"]
fig = gmaps.figure(center=(-20.0, 135.0), zoom_level=5.5)
marker_layer = gmaps.marker_layer(locations, info_box_content=hotel_info)
fig.add_layer(marker_layer)
# 11b. Display the figure
fig
```

(