1. Register or sign in to GCP (https://console.cloud.google.com/).
2. Download and install GCP SDK (https://cloud.google.com/sdk/install).
3. Enable Cloud Run API (https://console.cloud.google.com/apis/library/run.googleapis.com)
4. Google cloud shell
5. Create a project
6. Search for google cloud shell console
7. Open the terminal and enter the following
8. `Python3 -m venv .venv`
9. `. ./.venv/bin/activate`
10. `Add requirements.txt file to the editor:`

```
dash
Flask
Flask-Compress
itsdangerous
Jinja2
matplotlib
numpy
packaging
pandas
patsy
pipreqs
plotly
```

11. Copy the created dash .py file and paste inside the editor inside the GCP

12. Add the following command right after the my_app: Save the file as app.py

```
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import  Input, Output
import dash as dash
external_stylesheets =
['https://codepen.io/chriddyp/pen/bWLwgP.css']
my_app = dash.Dash('My app',
external_stylesheets=external_stylesheets)


server = my_app.server

my_app.layout = html.Div([
dcc.Slider(id='my-input',
   min = 0,
   max = 90,
   step= 1,
   value= 70,
),
html.Br(),
   dcc.Slider(id="second_slider",
         min=-10,
         max=35,
         step=.5,
         ),
])

@my_app.callback(
  Output(component_id='second_slider', component_property='value'),
  [Input(component_id='my-input', component_property='value')]
)
def update_reza(input):
```

```
  return (input-32)/1.8
if __name__ == '__main__':
    my_app.run_server(debug=True, host='0.0.0.0', port=8080)
```

13. Add the Docker file inside the editor- Name the file as ‘Dockerfile’.

```
# https://hub.docker.com/_/python
FROM python:3.8-slim-buster

# Copy local code to the container image.
ENV APP_HOME /app
ENV PYTHONUNBUFFERED True
WORKDIR $APP_HOME

# Install Python dependencies and Gunicorn
ADD requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt && pip install --no-cache-dir
gunicorn
RUN groupadd -r app && useradd -r -g app app

# Copy the rest of the codebase into the image
COPY --chown=app:app . ./
USER app

# Run the web service on container startup. Here we use the gunicorn
# webserver, with one worker process and 8 threads.
# For environments with multiple CPU cores, increase the number of workers
# to be equal to the cores available in Cloud Run.
CMD exec gunicorn --bind :$PORT --log-level info --workers 1 --threads 8 --timeout 0
app:server
```

14. Enable services through GCP terminal

gcloud services enable containerregistry.googleapis.com

15. Docker build

docker build -f Dockerfile -t gcr.io/covid-341822/test:test .

16. Docker push

docker push gcr.io/covid-341822/test:test

17. Docker deploy

gcloud run deploy dashapp --image gcr.io/covid-341822/test:test