### Cover:

Student Name: Kieran Mc Gowan

Email: kieran.mcgowan9@mail.dcu.ie

Student ID Number: 19214800

Programme: MCM - M.Sc. in Computing

Module Code: CA687

Lecturer: Long Cheng

Assignment Title: Assignment 1 - Cloud App Development.

Submission Date: Monday, 6th April 2020.

# 

# 

### Disclaimer:

An essay submitted to Dublin City University, School of Computing for module CA640 Professional and Research Skills in Computing.

I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy.

I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood [DCU Academic Integrity and Plagiarism Policy](http://www.dcu.ie/registry/examinations/index.shtml).

Name: Kieran Mc Gowan

Date: Monday, 6th April 2020.

## Cloud App Development.

## Background:

1. Use a cloud data processing technology we have learned about in this course.
2. Include a user interface (web/mobile) where users who give an input to the application can get a responsible output.
3. Use a large public dataset. For example, we can get a dataset from the following links:
4. The application should finally work in the cloud rather than locally.

## Introduction and Motivation for the App:

It’s Monday 6th April 2020, week 3 of an unprecedented global lockdown. COVID-19 has taken the world by surprise, causing massive levels of sickness and loss of life. There is no cure, no vaccine and no end in sight. We live in a globalized society, where people are free to travel anywhere at any time. These are ideal conditions for the transmission of a novel virus and it has been said that it was only a matter of time before the world faced a pandemic of the magnitude we are seeing today.

There is however some comfort to be taken in the data behind the virus. Information is available in Open Data repositories to report on where the virus first originated, the number of live infection cases, the daily increase rates and the subsequent deaths broken down by country and region throughout the world. Armed with this information we can analyze patterns, identify trends and ultimately predict when or if it is likely to subside.

The motivation of this app is to process the freely available raw data associated with the outbreak, cross-reference it with GEOIP reference points, and represent that data visually on publicly available heatmap. Building this app will inform the visitor of daily new infection hotspots while allowing me the time and space to learn and put into practice several new technologies learned throughout the year of study.

## Choice of Technologies:

* Google Cloud Platform
* Google Cloud Data Proc
* Apache Hive
* Apache Hadoop
* Linux Shell Scripting
* Git
* NodeJS
* Google Maps API

## Data Sources:

### COVID-19 Corona Virus data

#### Publisher: European Centre for Disease Prevention and Control

* ECDC is an EU agency aimed at strengthening Europe's defenses against infectious diseases. The core functions cover a wide spectrum of activities: surveillance, epidemic intelligence, response, scientific advice, microbiology, preparedness, public health training, international relations, health communication, and the scientific journal Eurosurveillance. data on the geographic distribution of COVID-19 cases worldwide

#### Data: The daily data on geographic distribution of COVID-19 cases worldwide

* <https://opendata.ecdc.europa.eu/covid19/casedistribution/csv>

### World Cities GEO locations

#### Publisher: SimpleMaps

* SimpleMaps have been developing, selling, and supporting interactive maps for over 9 years.
* Basic World Cities Database: The Provider offers a Basic World Cities Database free of charge. This database is licensed under the Creative Commons Attribution 4.0 license as described at: https://creativecommons.org/licenses/by/4.0/.

#### Data: Basic World data data

* <https://simplemaps.com/static/data/world-cities/basic/simplemaps_worldcities_basicv1.6.zip>

## Data Processing:

The aim of this project is fully automated no-touch daily updating heatmap. This is achieved with the use of several technologies all working together. The data flow looks something like this.

Hive Setup:

1. Create a new DataProc cluster on Google Cloud
2. SSH to the master cluster
3. Git clone the project repos
4. Load the city GEOP location data into a Hive table

Hive Cronjob daily:

1. Backup yesterdays data
2. Download the latest covid-19 cases data
3. Load this data into a hive table
4. Execute an INNER JOIN across the two tables to join country based COVID-19 statistics with country based GEO location co-ordinates.
5. Commit the new report data to the github repo

Web server daily:

1. Git pull the latest report from the repository
2. Restart the web app to view the latest heatmap.

## Challenges and Lessons Learned:

The biggest challenge with this project was to get comfortable using Apache Hive. As a software developer, I am familiar with relational databases such as Oracle and MySQL so I thought moving to Hive would be an easy transition. However, I found the learning curve to be quite high and the feedback to the terminal client to be not very user friendly.

For example in several cases when running Hive queries I caused the Hive interpreter to hang for long periods, and I had no idea why it was hanging. This was particularly the case during JOIN queries where it was calculating a cartesian product. Sometimes the query would execute and sometimes it would just hang, the only solution being to restart the cluster.

This project helped me to appreciate the need for Hive, that being a more intuitive interface to Hadoop Mapreduce, that creates Map-reduce jobs from the interpreted SQL like queries and then farms off the jobs to the Hadoop nodes. This part of the project was actually quite fun, interesting and I learned a lot from it.

It would be nice to spend a little more time making the map more user friendly, and I’m aware that the map is not currently processing the weights correctly.

## Demo.

### Sourcecode:

#### <https://github.com/sligokid/dcu-covid-19-hive>

### Live Endpoint:

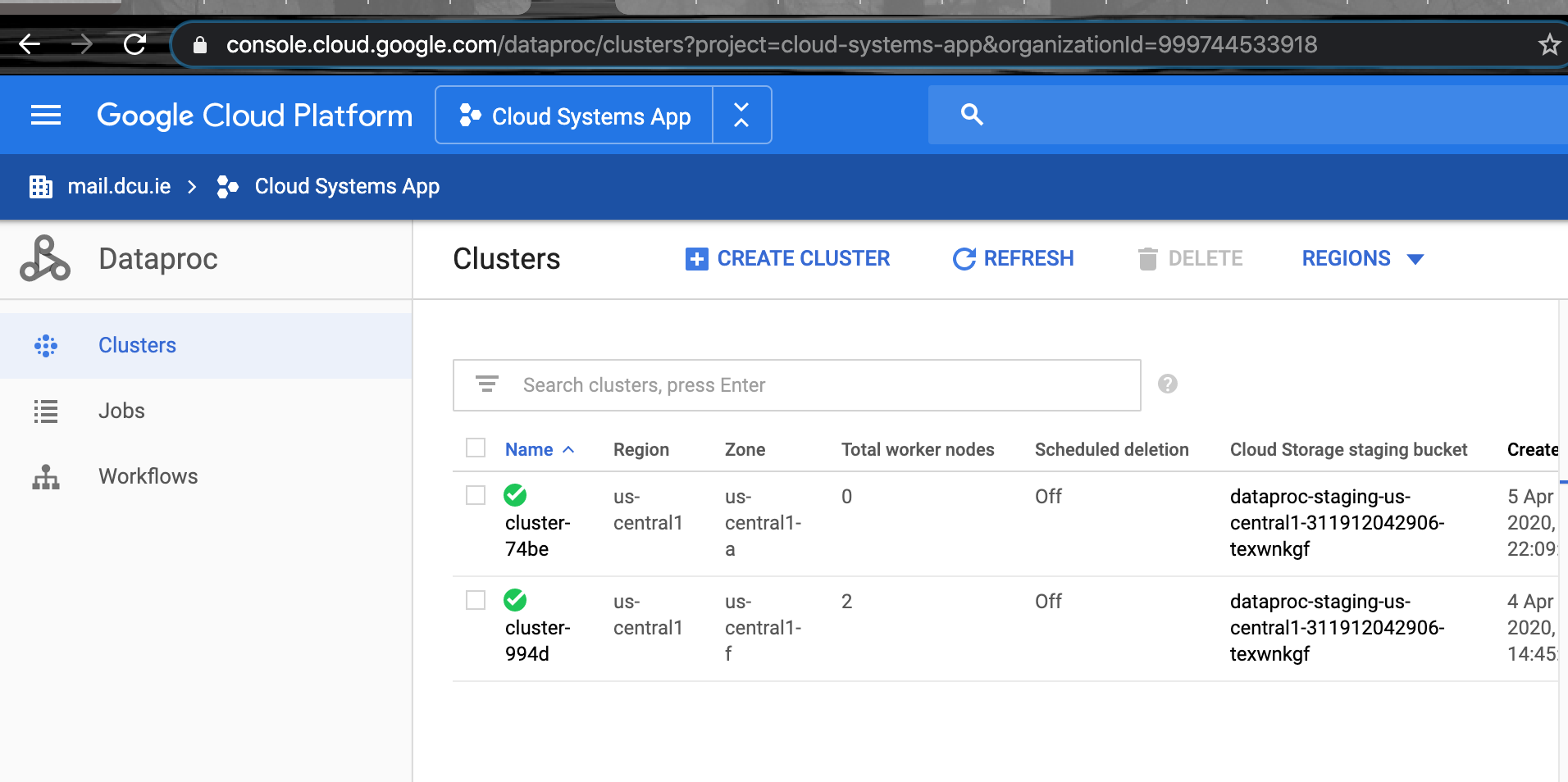
#### <http://34.66.242.221:8080/>

### Demo Screencast:

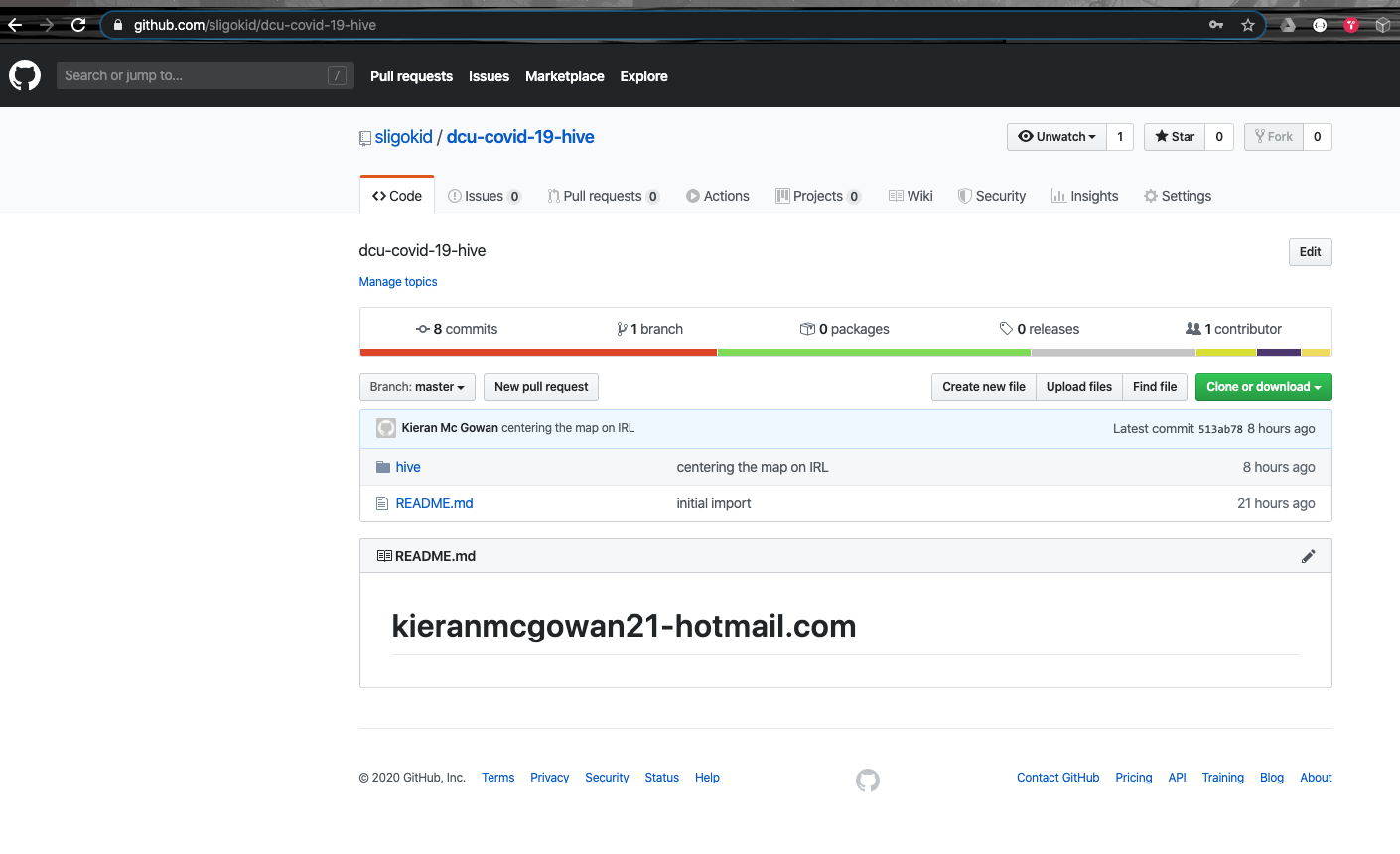
#### <https://youtu.be/ubysPIUo2O8>

## Appendix.

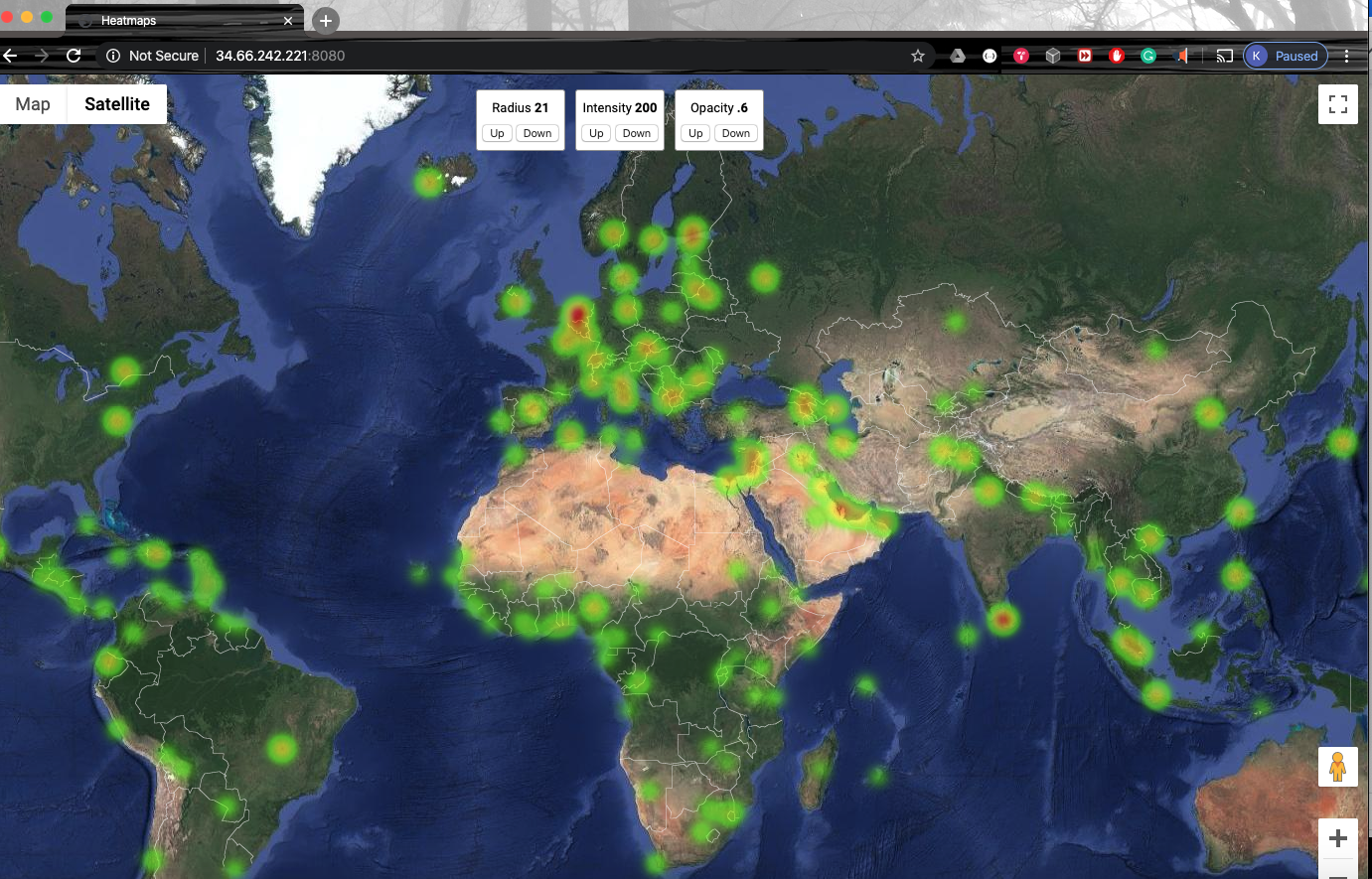
### GCP Clusters:



### Github:



### App:



### 

### 