

# Cluster and Cloud Computing Assignment 2

## City Analytic on the Cloud

Zhaofeng Qiu 1101584

University of Melbourne — May 25, 2020

### 1 System Design and Architecture

The architecture of our system is in well designed, which is aimed at providing both a beautiful front-end web application and a RestFul API server with high availability, high scalability and fault tolerance. The overall architecture is shown in Figure 1.

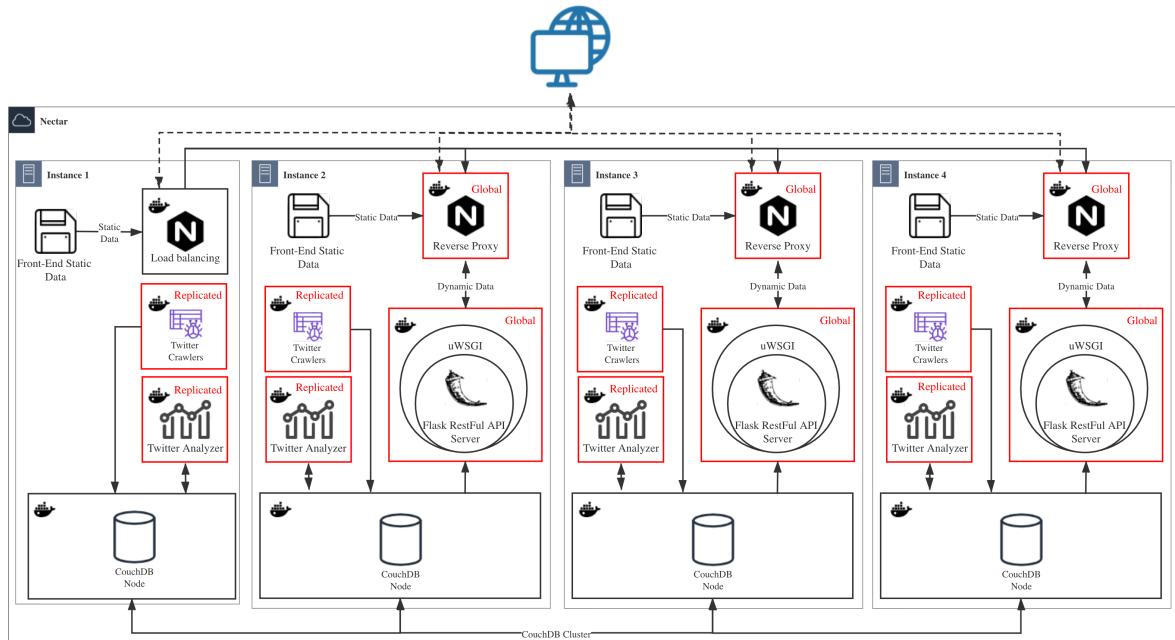


Figure 1: Velocity Re-estimation

#### 1.1 High Availability

Through in-depth thinking and repeated practice, we continue to optimize our architecture to achieve a high availability with limited resources. We mainly use the following method to improve the availability of our system.

Firstly, we run a NginX server in our first instance to provide Load balancing. For requests that need to access the dynamic data in our database, it would pass the request to the other three server averagely. In this way, we can efficiently distribute the incoming network traffic to our backend servers in the cluster. When it comes to the situation that the first instance may crash, users can still access our application using the IPs of the other three nodes. But this solution is not good enough and is contradict to high availability. We had tried to use two NginX server with a software called Keepalived to provide a more fault-tolerant system. However, we found out that we are not allow to use floating IP in Nectar. Since we cannot set a

virtual ip for our NginX servers, the only way to handle this issue is to let the other three nodes to keep the full functionality of our application. User can access our application through any IP of our instances, but only the IP of the first instance would provide load banlancing control.

Secondly, we use docker swarm to manage part of our services. We set three swarm managers in our cluster. Managers in swarm

## **1.2 Dynamic and Static Separation**

When the first instance is alived, we

## **References**