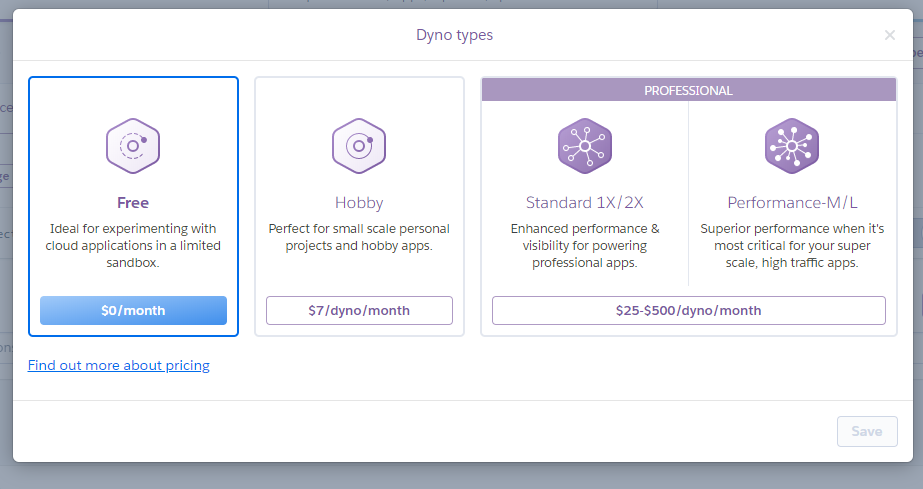
* How is your project architecture related to the theory taught in the lecture?

The project uses a PaaS service like YouTube API from google API and Heroku. Also use a data storage service from Redis. Heroku and Redis are distributed systems, they have a different server on different locations, but our service is not distributed. Heroku App is built on a virtual machine called slug and the instance launched from this virtual machine is called dyno. It belongs to instruction set architecture level virtual machine. We also acted as a cloud service consumer that interact with a cloud service’s application programming interface (API) like Tian API, Imgur API and YouTube API.

* Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

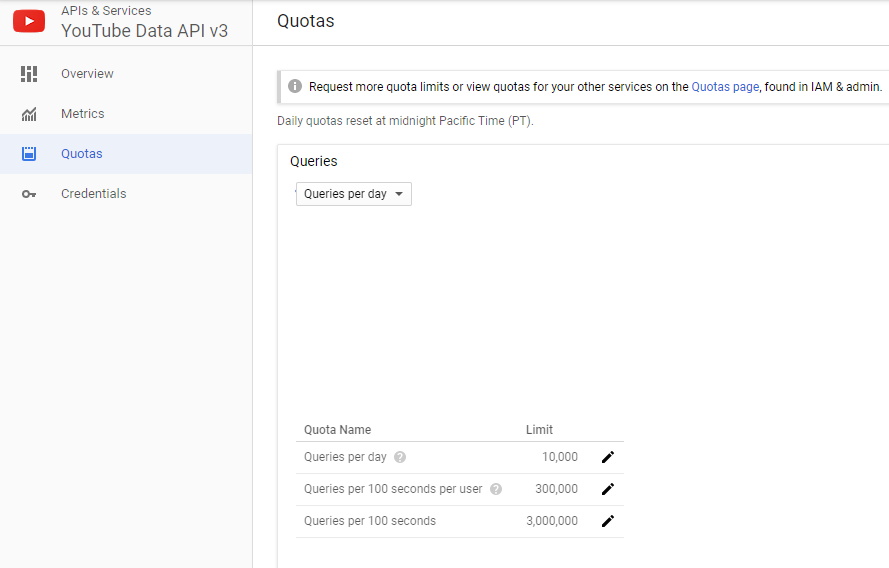
In order to fix the multiple-users problem. Physical ways are to use a better and closer Heroku server and Redis server, it can increase the handling speed.

Or connect to more servers to handle multi-millions of users to use simultaneously. Heroku apps can be scaled to run on multiple dynos simultaneously. By buying upgraded services.

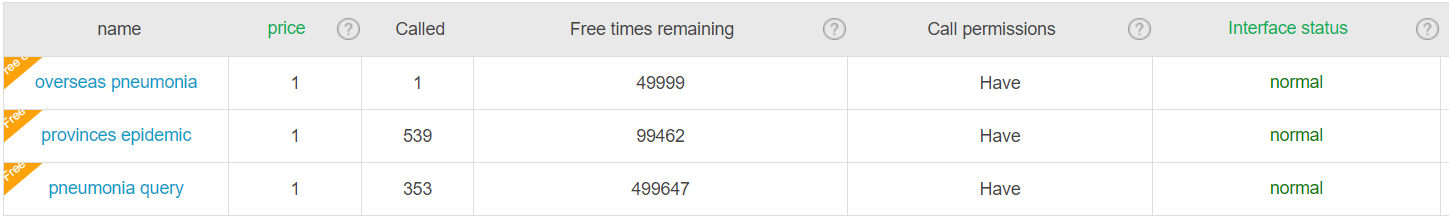


Also, the YouTube API query is limited per day. Too many users may reach the limit quickly.

So, we need to buy more powerful services form google to scale up the capacity.



Similarly, we can recharge at tianapi.com to get more API calls to satisfy a large number of requests from users.



Apart from deploying the chat bot in Heroku, we can deploy it with Google Cloud service.

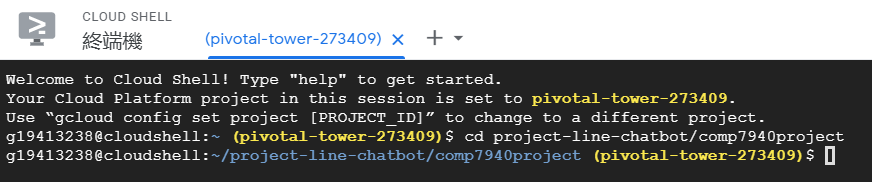
Here is how to increase the capacity of chatbots through Google App Engine, not including deploying the chat bot to the Google Cloud.

1. Use the Cloud Shell to clone and browse to the "project" code.

Enter the following command in the Cloud Shell:

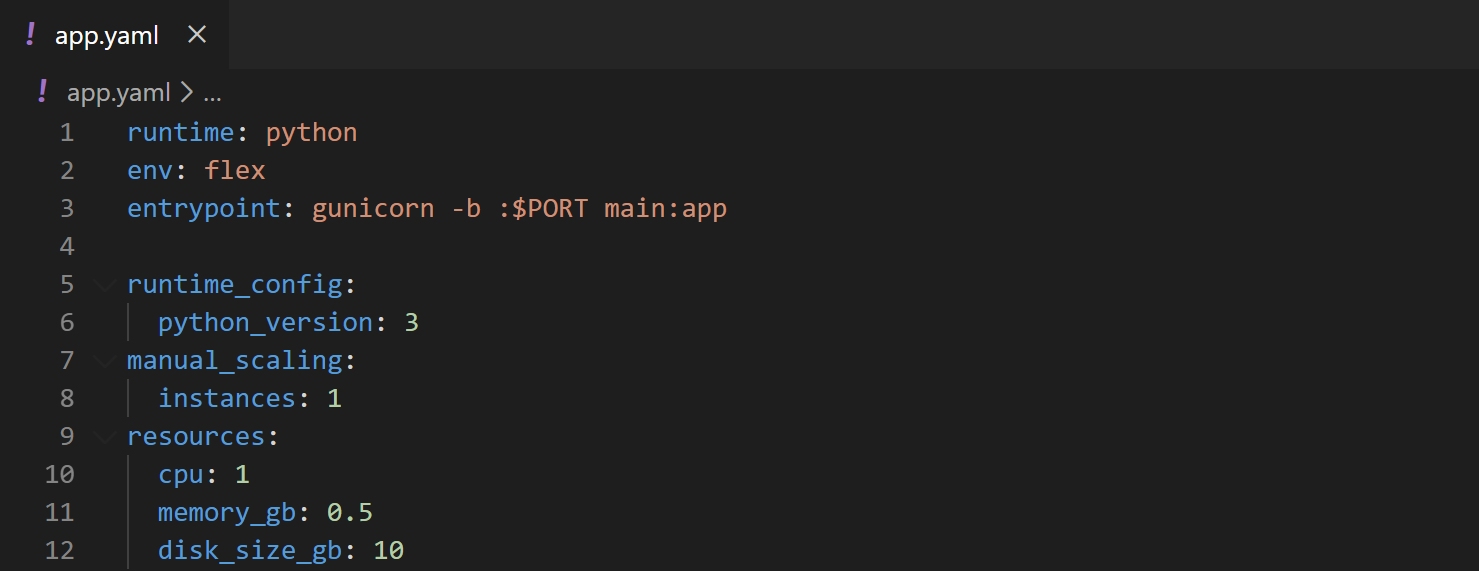
git clone <https://github.com/singsing215/project-line-chatbot.git>

cd project-line-chatbot/comp7940project



1. Create app.yaml file

Python applications in App Engine are configured using the app.yaml file. This file contains other general Settings such as CPU, memory, network and disk resources, expansion, and environment variables.

app.yaml code:

This file contains 1 CPU, 0.5GB of memory, and 10GB of disk resources.

1. Deploy to App Engine

Create an application:

gcloud app create

Deploy using the Cloud Shell:

gcloud app deploy app.yaml --project pivotal-tower-273409

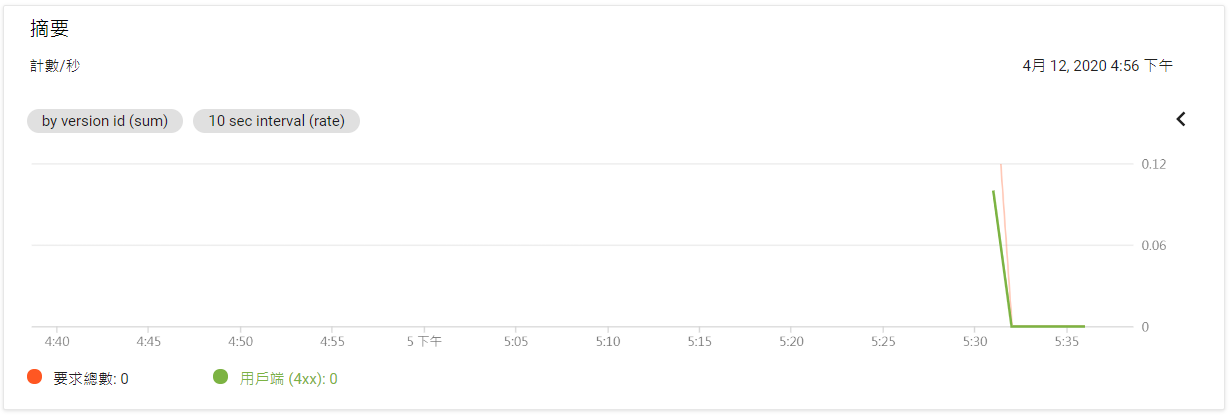
The application is deployed, and the default address is the subdomain on appspot.com, starting with my project ID: pivotal-tower-273409.appspot.com



1. View the application status

We can monitor the status of your application from the App Engine information home page:





With the above deployment, we increase capacity of 1 CPU, 0.5GB of memory, and 10GB of disk resources to the chatbot. If we want to adjust the CPU, memory, or disk of application, we can modify the parameters of app.yaml file.

All in all, we can upgrade the Heroku service, improve the request quota of YouTube API and Tian API, or deploy another cloud service like Google App Engine to increase capacity of our chat bot service.

* Can you identify if you bot is one of the examples of PaaS, IaaS, SaaS? Explain your answer.

Heroku is a PaaS platform, and LINE bot is implemented on the Heroku platform, so our bot is the examples of PaaS. Actually, our Heroku service also subscribed other service, including Redis, Tian API, Imgur API and YouTube API, which makes our Heroku service as SaaS.

Our line chatbot is SaaS, people can connect the bot so he can directly use the function like getting the latest infected number in each city/receive particular YouTube video. Customer just have limit control. Service only provisioned by us.