

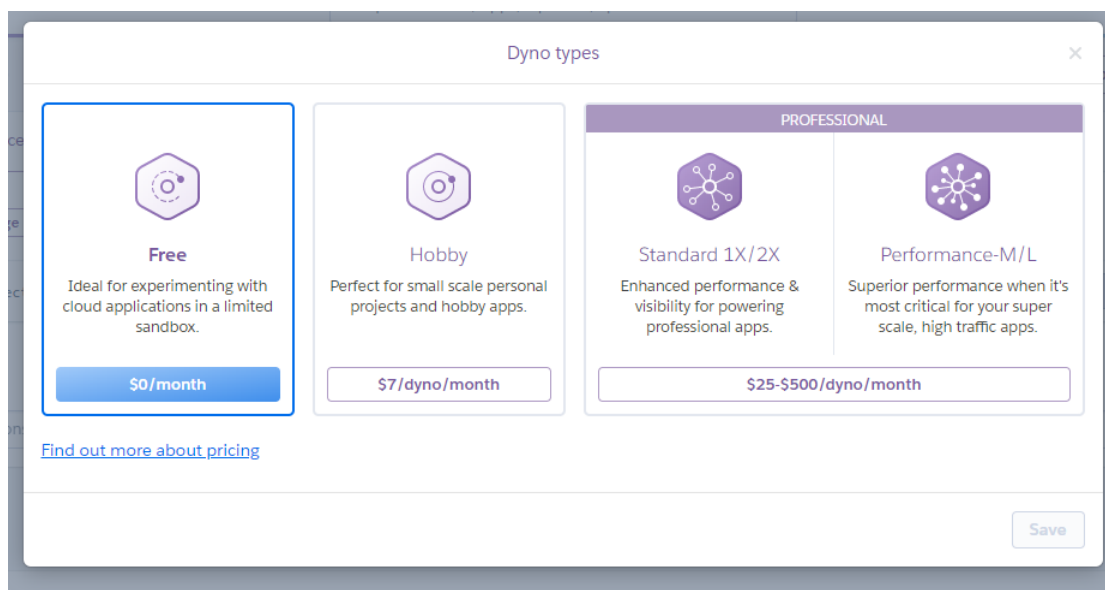
- How is your project architecture related to the theory taught in the lecture?

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 from Google to scale up the capacity.

APIs & Services

YouTube Data API v3

Overview

Metrics

Quotas

Credentials




Quotas

Request more quota limits or view quotas for your other services on the [Quotas page](#), found in IAM & admin.

Daily quotas reset at midnight Pacific Time (PT).

Queries

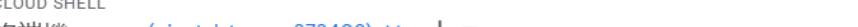
Queries per day ▾

Quota Name	Limit	
Queries per day [?]	10,000	
Queries per 100 seconds per user [?]	300,000	
Queries per 100 seconds	3,000,000	

name	price	Called	Free times remaining	Call permissions	Interface status
overseas pneumonia	1	1	49999	Have	normal
provinces epidemic	1	539	99462	Have	normal
pneumonia query	1	353	499647	Have	normal

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

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



A terminal window titled "CLOUD SHELL" and "終端機". The window shows a welcome message and instructions for using the Cloud Platform project. The user's current directory is `~/project-line-chatbot/comp7940project`. The terminal output is as follows:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to pivotal-tower-273409.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
g19413238@cloudshell:~ (pivotal-tower-273409)$ cd project-line-chatbot/comp7940project
g19413238@cloudshell:~/project-line-chatbot/comp7940project (pivotal-tower-273409)$
```

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:

```
! app.yaml ×
! app.yaml > ...
1 runtime: python
2 env: flex
3 entrypoint: gunicorn -b :$PORT main:app
4
5 runtime_config:
6   python_version: 3
7 manual_scaling:
8   instances: 1
9 resources:
10  cpu: 1
11  memory_gb: 0.5
12  disk_size_gb: 10
```

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

3. 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

```
target service: [default]
target version: [20200412t172627]
target url:      [https://pivotal-tower-273409.df.r.appspot.com]
```

4. View the application status

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

帳單狀態

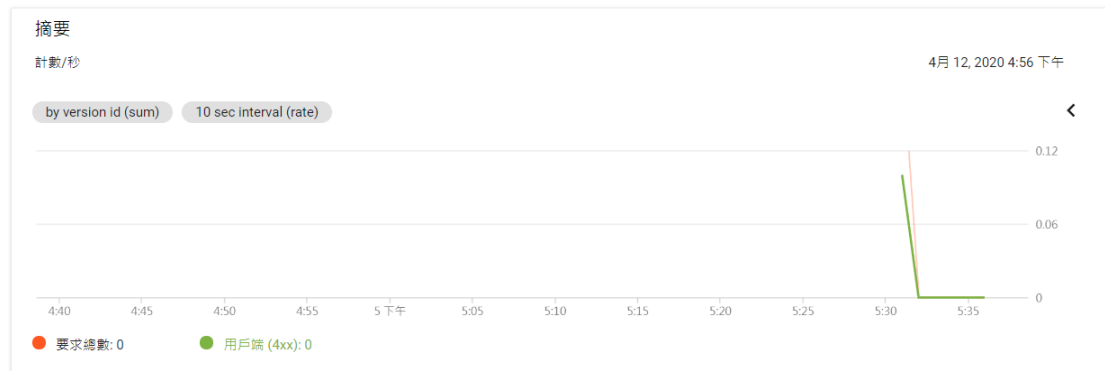
已啟用
配額每隔 24 小時會重設一次。下次重設時間：22 小時後

資源	使用情況	可計費	價格	費用
前端執行個體時數	0.01 個執行個體小時	0.00	\$0.063/小時	\$0.00
運出頻寬	0.000001 GB	0.00	\$0.12/GB	\$0.00
Cloud Storage B 級作業	0	0	\$0.40/百萬次操作	\$0.00
Cloud Storage A 級作業	0	0	\$5.00/百萬次操作	\$0.00
最近 2 小時的預估費用				\$0.00*

* 根據預估值的計算方式，個別資源費用的總和可能與顯示的總值不一致。

目前負載

URI	要求/分鐘 目前	要求 過去 24 小時	執行階段 MCycles 過去 1 小時	平均延遲時間 過去 1 小時	錯誤項目 過去 24 小時
/favicon.ico	0.2	1	9	5 毫秒	查看追蹤
/	0.2	1	3,071	1,534 毫秒	查看追蹤



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.