

COMP7940 Cloud Computing
Group project : Line bot
Milestones 4

Students Name & Student ID

Chan Chun Kuen (19459076)
Kwok Man Hin (19411081)
Yung Wah Shun (19449992)

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

The line chatbot is Client-Server Model that partitions tasks or workloads between providers of a service and service requesters.

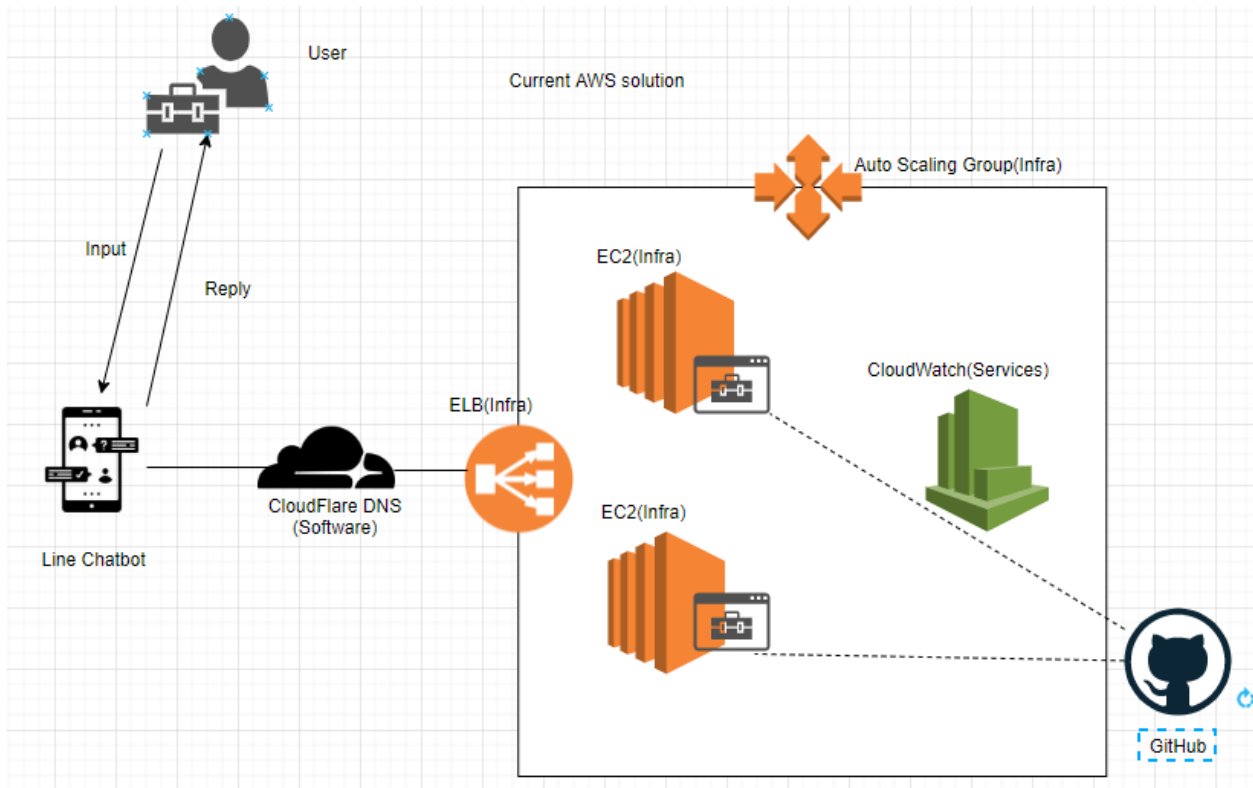
For communication paradigms, the line chatbot is Indirect Communication via a third party. It is space uncoupling as the senders do not need to know who they are sending to and it is time uncoupling as the senders and receivers do not need to exist at the same time. There are message queues that producer processes can send messages to a specified queue, and consumer processes can receive messages from the queue.

For the architectural patterns, it is three-tier architecture. There is a one-to-one mapping from logical elements to physical elements. Each tier has a well-defined role. Tier 1 is mobile device and personal computer for user view and control the line chatbot. Tier 2 is application server for the line chatbot application logic. Tier 3 is database server for database manager. The application is the line messenger itself. In mobile version of line chatbot, Apple IOS and Google Android are used as operating system; while in desktop version of line chatbot, Microsoft Windows and Apple macOS are used as operating system.

It is an asynchronous distributed system as there is no bound on process execution speeds, message transmission delays and clock drift rates. Omission failure may occur when a process or communication channel fails to perform pre-defined line chatbot actions. Process Omission Failure exists when a process is halted and does not execute any further

steps of its program. Other processes cannot detect the failure, but only indicate that the suspicious process is not responding. Communication Omission Failure exist if it does not transport a message from outgoing message buffer to incoming message buffer. This is known as dropping messages which is generally caused by a lack of buffer space at the receiver or at an intervening gateway, or by a network transmission error.

Hardware architecture



Heroku Back Up Plan

Heroku



PostgreSQL



Redis



Python

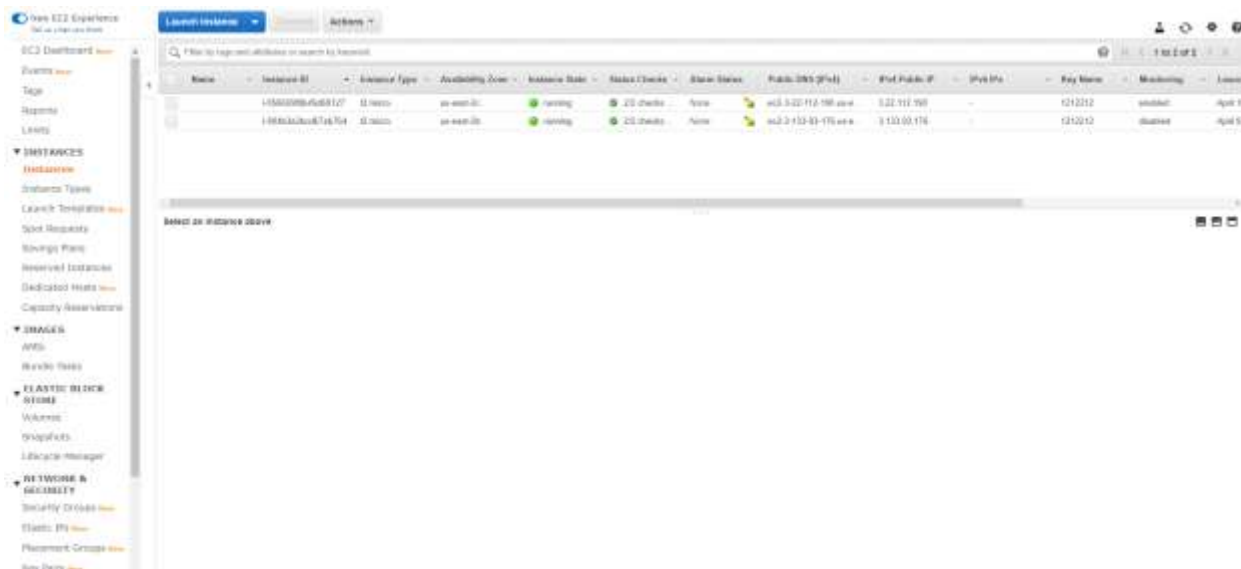


- 2) Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

Heroku was used in the beginning of our development and AWS Elastic Compute Cloud (EC2) was used as the final cloud storage devices to provide fixed-increment capacity allocation in support of the pay-per-use mechanism. Both could be used to edit, monitor, deploy and scale our Line chat bot.

Since we may not increase the capacity in Heroku freely, the Line chat bot is transferred to AWS Elastic Compute Cloud (EC2) for larger capacity of the chat bot service. There is also no charge for using EC2 service. These include the EC2 instances that comprise different combinations of CPU, memory, storage, and networking capacity, S3 storage, and so on. As a trial version, all new users can opt for a free usage tier to deploy a low-traffic app.

Combining Elastic Load Balancers with EC2 Auto Scaling helps to manage and control the AWS workloads. The main function of an Elastic Load Balancer, commonly referred to as an ELB, is to help manage and control the flow of inbound requests to a group of targets by distributing these requests evenly across the targeted resource group. These targets could be a fleet of EC2 instances, AWS Lambda functions, a range of IP addresses, or even containers. The targets defined within the ELB could be situated across different availability zones (AZs) for additional resilience or all placed within a single AZ



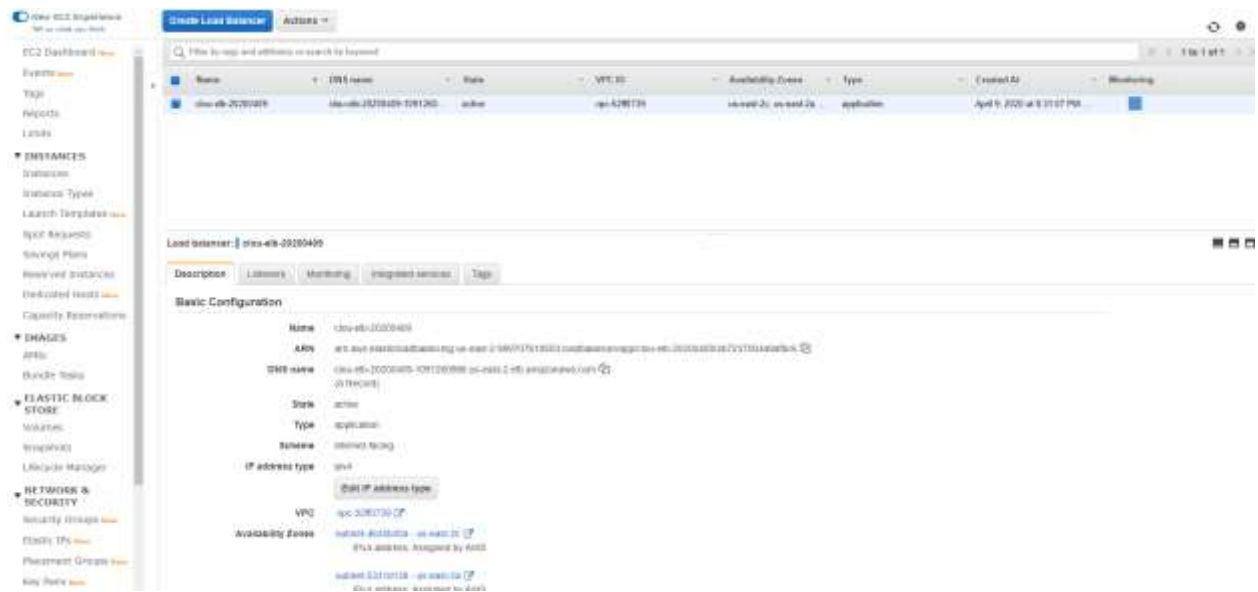
Elastic Load Balancers (Proof)

To solve the problem of sudden spike in traffic, the AWS Elastic Load Balancer acts as the point for receiving incoming traffic from users and evenly distribute the traffic across a greater number of instances. By default, the ELB is highly available since it is an AWS managed service, which works to ensure resilience. Although it might seem the ELB is a single point of failure, the ELB is in fact comprised of multiple instances managed by AWS. One of the many advantages of using an ELB is the fact that it is managed by AWS and it is, by definition, elastic. This means that it will automatically scale to meet your incoming traffic as the incoming traffic scales both up and down.

```
*** router.lan can't find covid19bot.khomecloud.site: Non-existent domain

C:\Users\KennyAdmin>nslookup covidbot.khomecloud.site
Server: router.lan
Address: 192.168.88.1

Non-authoritative answer:
Name: clou-elb-20200409-1091260986.us-east-2.elb.amazonaws.com
Addresses: 3.135.82.142
           3.20.200.179
Aliases: covidbot.khomecloud.site
```



EC2 Auto Scaling



Create an Auto Scaling group

operating

Filter conditions:

1 to 1 of 1 Auto Scaling group

name	Launch configuration	Example	Request	Min	max	Availability zone	Default cooling time	Grace period for b
asp-targetconfg	targetconfg		2	2	2	us-east-2a, us-east-2b, us-e	300	300

Auto Scaling group: asp-targetconfg

details

activity history

expansion strategy

examples

monitor

tags

placement operations

life cycle hook

Auto strategy

targetrestriction

operating

Strategy type:

Target tracking extension

Time to execute strategy:

Required to maintain the application load balancer request count of each target at 10000

Please do the following:

Add or delete instances as needed

Instance needs:

300 Warm up after expansion in seconds

Disable reduction:

no

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

For the beginning of the project, Heroku was used, which is a *Platform as a Service* (PaaS) product. This platform equips us with a ready runtime environment and application servers. Users could be released from tasks related to servers, virtualization, storage, and networking, since the provider is responsible for them. Also, users as app creators did not need to worry about operating systems, middleware, software updates, etc.

For enhancing the capacity of the chat bot, Amazon's Elastic Compute Cloud (EC2) is used, which is the legitimate IaaS (Infrastructure As A Service) platform. And the provider will supply its users with the complete infrastructure such as virtual networks, virtual machines and much more. It detailed delving into the configuration of server infrastructure, adding database instances, and other activities related to app deployment.

