Milestone4

Group2

Cheong Chin Wang 19482442

CHENG Chen 19413122

CHEN Kun 19426305

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

From the lecture, we know that cloud computing is a kind of ‘out-sourcing’ in which we employ service from others to achieve better reliability, economic benefit, etc.

In our project, we make use of different cloud services to provide our service.

From the hardware perspective, the codes are run on the Heroku server, whether than on our computer to achieve higher reliability and lower cost(e.g., management cost). Also, some messages are stored in the Redis server, which is located in another place.

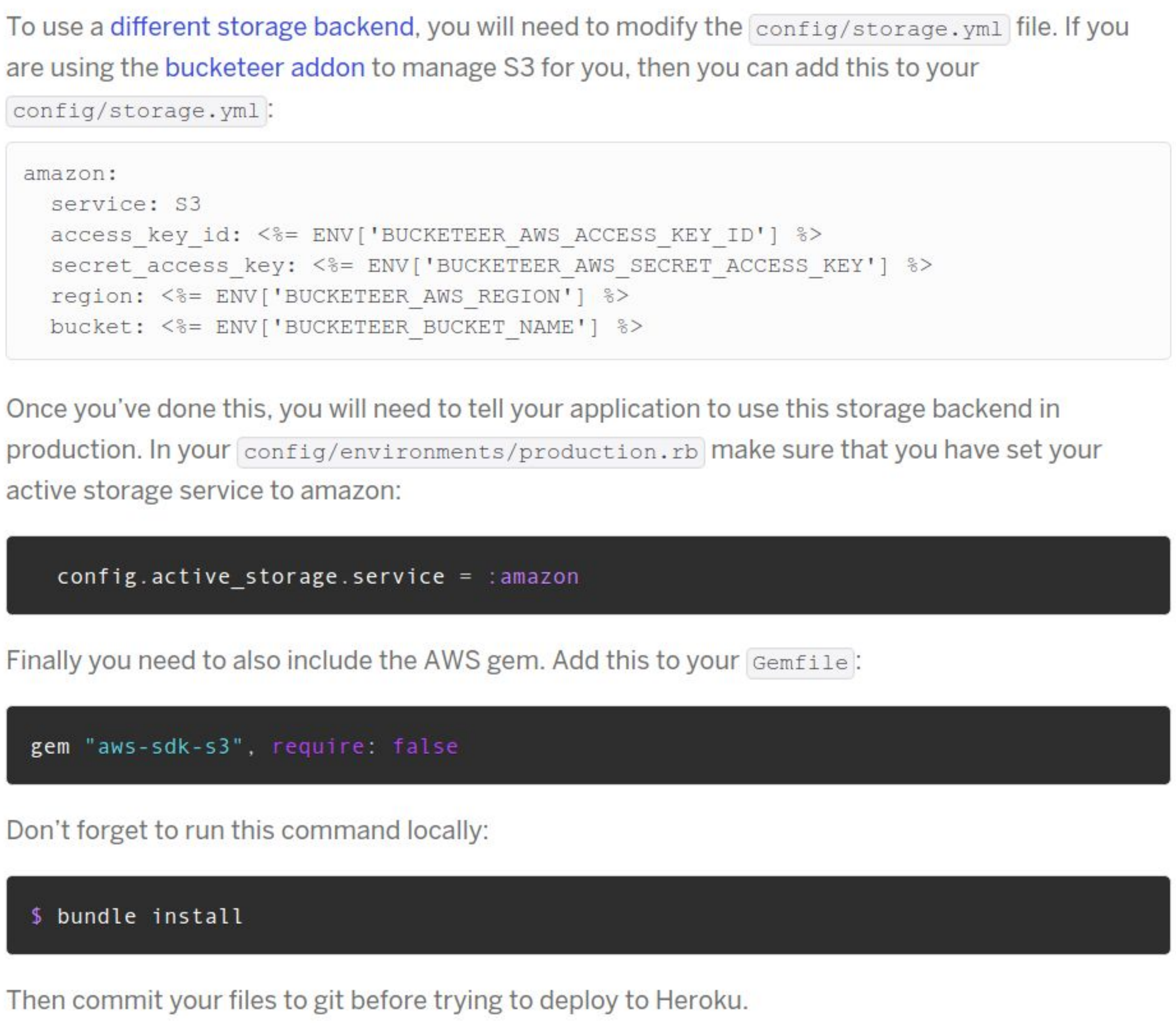
In the software perspective, we adopt RPC (Remote procedure calls)(procedures in processes on remote computers can be called as if they are local) to call external services such as google cloud translate and Redis service.

In terms of architectural styles, our line bot adopts the Client-Server model in which the client(Line mobile app) calls chatbot service from the Heroku server.

2. Can you demonstrate, with some screen cap, how to increase the capacity of your chatbot service?

When we need more space for storing information such as news or FAQ, the limited space provided by Heroku will be an issue.
To scale up the space capacity, we can try to replace the ephemeral disk of Heroku by some third party storage system. For example, we may try Amazon’s s3.

To do so,



To use a different storage backend, you will need to modify the `config/storage.yml` file. If you are using the bucketeer addon to manage S3 for you, then you can add this to your `config/storage.yml`:

```
amazon:
  service: S3
  access_key_id: <%= ENV['BUCKETEER_AWS_ACCESS_KEY_ID'] %>
  secret_access_key: <%= ENV['BUCKETEER_AWS_SECRET_ACCESS_KEY'] %>
  region: <%= ENV['BUCKETEER_AWS_REGION'] %>
  bucket: <%= ENV['BUCKETEER_BUCKET_NAME'] %>
```

Once you've done this, you will need to tell your application to use this storage backend in production. In your `config/environments/production.rb` make sure that you have set your active storage service to amazon:

```
  config.active_storage.service = :amazon
```

Finally you need to also include the AWS gem. Add this to your `Gemfile`:

```
gem "aws-sdk-s3", require: false
```

Don't forget to run this command locally:

```
$ bundle install
```

Then commit your files to git before trying to deploy to Heroku.

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

Our chatbot is an example of SaaS(Software-as-a-Service). We provide a software service to our users on the front end.
In our case, users have very limited administrative control, meaning that they can only access services provided by the chatbot.

But we still use other non-SaaS services. For example, we use Heroku, which is a PaaS(Platform-as-a-Service) for running our bot. As Heroku provides a platform for hosting our app. Notice that Heroku also uses Amazon Web Service (AWS), an Infrastructure as a Service (IaaS) as basis.

On the other hands, we exploit Redis, which is a storage-as-a-service for remote storage.