

Assignment 2 - Cloud Computing
RMIT University

Social Media Platform

Students: Clinton Pham (s3605044), Sean Tan (s3806690)

TABLE OF CONTENTS

| | |
|--------------------------------------|-----------|
| Signed Contribution Agreement | 2 |
| Links | 2 |
| Summary | 3 |
| Introduction | 3 |
| Related Work | 4 |
| Software Design/Architecture | 4 |
| Implementation | 7 |
| AWS Services Used | 17 |
| A small user manual | 21 |
| References | 26 |
| Video (optional) | 26 |

a. Signed Contribution Agreement

| | |
|--|---|
| Student Name: Sean Tan | Student Name: Clinton Pham 30/01/2021 |
| Student ID: S3806690 | Student ID: S3605044 |
| Contributions: 1. AWS Services: DynamoDB, S3 Bucket, Elastic Beanstalk, AWS Cloudfront, AWS Translate 2. Code: Implemented backend functions related to the AWS Services mentioned above and some frontend designs 3. Report: Implementation section of report | Contributions: 1. AWS Services: AWS Cloud9, AWS SNS, AWS SES, AWS CloudWatch 2. Code: Implemented backend functions related to the AWS SES and AWS SNS, done most of the frontend design 3. Report: Done majority of the report |
| Contribution Percentage: 50% | Contribution Percentage: 50% |
| <i>By signing below, I certify all information is true and correct to the best of my knowledge.</i> Signature:  Date: 30/01/2021 | <i>By signing below, I certify all information is true and correct to the best of my knowledge.</i> Signature:  Date: 30/01/2021 |

b. Links

- V9-env.eba-nhm3yvem.ap-southeast-2.elasticbeanstalk.com
(Elastic beanstalk instance is not always online but can be put on if requested)

c. Summary

The purpose of this project is to allow users to connect to each other based on their personality description on their profile. Our goal is to enable family and friends to stay connected with each other and allow them to share updated posts to keep each other up to date. Additionally, we have implemented a cryptocurrency display section on our platform which displays the user's favourite cryptocurrency. Our goal with this implementation is to raise awareness for cryptocurrency.

d. Introduction

The use of social media has skyrocketed over the past two decades after the invention of the world wide web. For many people, the impact of social media has gone from being an entertainment platform to a fully integrated part of nearly every aspect of their daily lives.

Thus, we have decided to create a social media platform that displays only the most basic information about an individual to allow people to connect to each other if they share the same interest.

Our platform is different to the existing major social media platforms such as Facebook and Instagram as it also displays the users favourite **cryptocurrency** on their profile page. Our goal is to raise awareness for cryptocurrency as we're both firm believers in cryptocurrency. With the recent global pandemic of coronavirus, also known as COVID-19, it has accelerated the interest of digital currency which has prompted us to integrate cryptocurrency into our social platform.

In order to use our platform, users must have a registered account. Once users sign up for an account, they will have to enter information regarding their name, email address, about me, favourite cryptocurrencies and favourite websites. Registered users are able to search for friends and family on the platform, post on each other's wall, connect and chat with one another, see real-time prices of their three favourite cryptocurrencies and as an added bonus feature - transcribe recording audio file to text format.

Our platform enables users to build relationships with one another. Strong connections with friends, family, colleagues and the wider community provides happiness, security, support and a sense of purpose. Being connected with each other not only promotes healthy mental wellbeing but can also act as a barrier against mental problems such as anxiety and depression.

e. Related Work

Facebook

Facebook is a social media platform that gives people the ability to connect with their friends, family and the wider communities as well as businesses all over the world.

We got inspiration from facebook to create our social media platform. After exploring our options for assignment two, we decided it would be best to create a social media platform as it will have a lot of different services we can use to integrate from AWS. It will also allow us to enhance our backend as well as our frontend skills.

f. Software Design/Architecture

Table: Users

| Attribute Name | Value Type | Description |
|--------------------------|------------|--|
| username (partition key) | String | Unique username of a user used for login |
| User_type (sort key) | String | Type of user |
| fullname | String | Full name of the user |
| password | String | Password of a user used for login |
| email | String | Email of the user |
| phone | String | Phone number of a user, used for reset password feature |
| aboutme | String | Some info a user would like to display on their profile page |
| crypto1 | String | 1st favourite cryptocurrency of the user |
| crypto2 | String | 2nd favourite cryptocurrency of the user |
| crypto3 | String | 3rd favourite cryptocurrency of the user |
| website1 | String | 1st favourite website of the user |
| website2 | String | 2nd favourite website of the user |

| | | |
|----------|--------|-----------------------------------|
| website3 | String | 3rd favourite website of the user |
|----------|--------|-----------------------------------|

Table: FriendList

| Attribute Name | Value Type | Description |
|--------------------------|------------|---|
| Username (partition key) | String | Username of the selected user |
| Friendname (sort key) | String | Username of another user who is a friend/connected with the selected user |

Table: FriendRequest

| Attribute Name | Value Type | Description |
|--------------------------|------------|---|
| Username (partition key) | String | Username of the selected user |
| targetname (sort key) | String | Username of a user whom the selected user has sent a connect request to |

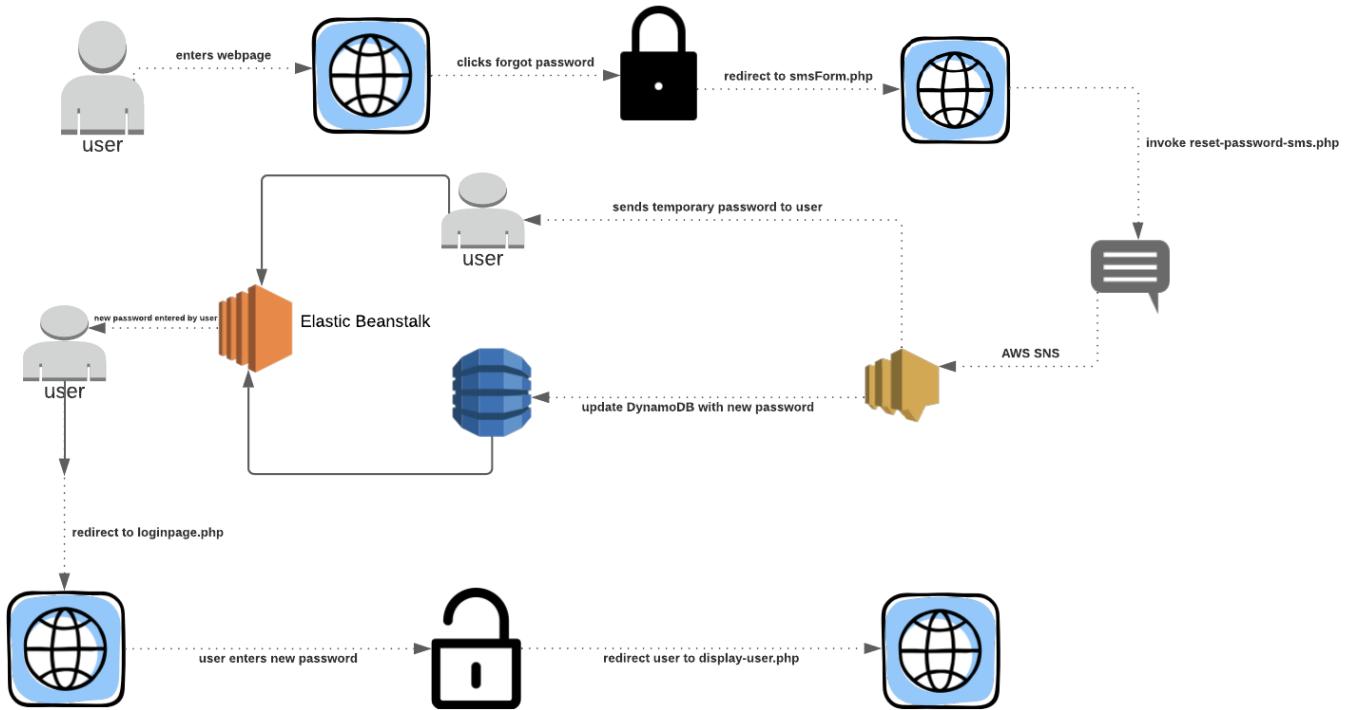
Table: Messages

| Attribute Name | Value Type | Description |
|----------------------------|------------|---------------------------------------|
| Friendpair (Partition key) | String | Unique 2 username combination key |
| Timestamp (Sort key) | String | Time and date the message was sent |
| author | String | Username of the author of the message |
| content | String | Content of the message |

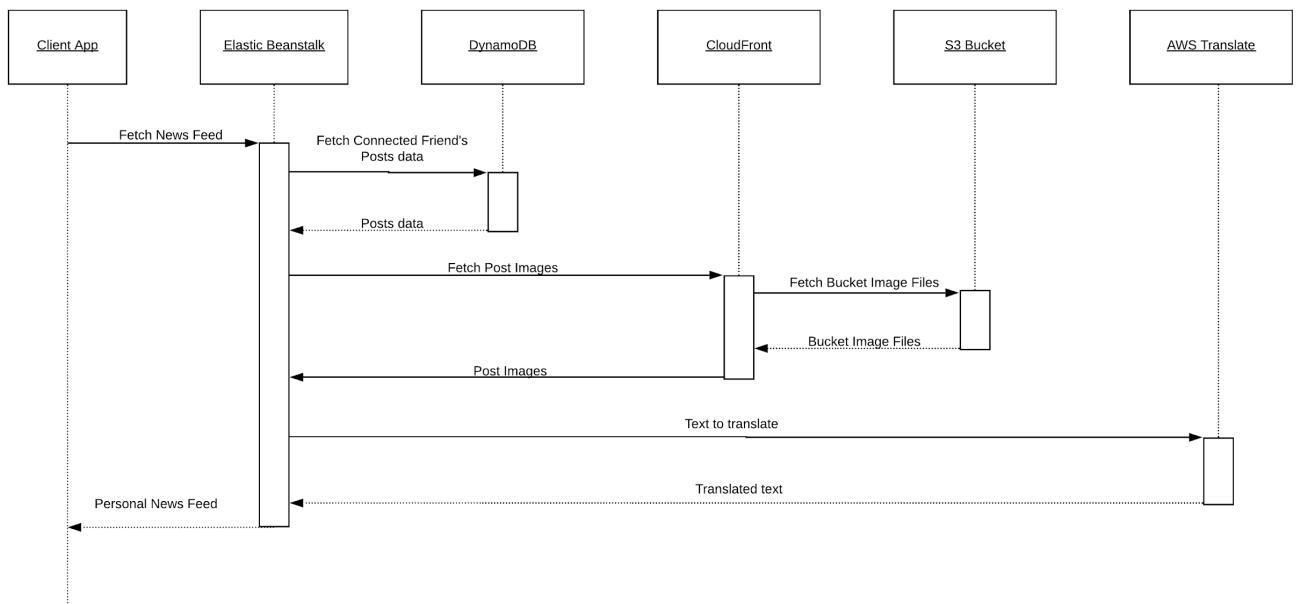
Table: Posts

| Attribute Name | Value Type | Description |
|--------------------------|------------|---|
| Username (Partition key) | String | Username of the user who created the post |
| timestamp | String | Time and date the post was created |
| content | String | Content of the post |
| imageURL | String | Link of the image used on the post which is stored in S3 bucket |
| language | String | Language the post is written in |

AWS SNS



News Feed feature's interaction with AWS Cloud Services



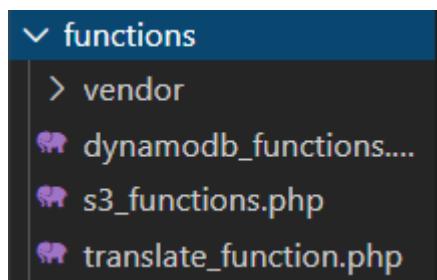
g. Implementation

Setup

Begin by ensuring AWS SDK for PHP is installed as a dependency via Composer which can be downloaded from the following link

- https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/getting-started_installation.html

Secondly, create a folder named 'functions' and inside create files named dynamodb_functions.php, s3_functions.php and translate_function.php where each is a class containing the client connecting to AWS services and the functions for the project. Majority of the features listed below will be using one or multiple functions contained in these classes.



Code for each file are as below:

DynamoDBFunctions class: <https://pastebin.com/9WmpcFsj>

S3Functions class: <https://pastebin.com/1iWLs10s>

AwsTranslateFunctions class: <https://pastebin.com/biDLQ0U9>

User

Login Page

Create the '*loginpage.php*' file with the code below in the root directory, the POST form in this page will call the Login() function in DynamodbFunctions class to authenticate the login, if successful user is directed to their profile page

- *Loginpage.php*: <https://pastebin.com/f1LP93X8>

Register Page

Create the registerpage.php file with the code below in the root directory, the POST form in this page will call the Register() function in DynamodbFunctions class to create a new User entity in DynamoDB table, if successful a success message will be displayed, if username is taken an error message will show

- *registerpage.php*: <https://pastebin.com/8NSYysLx>

Reset password using AWS SNS service

Create the reset_password_page.php file with the code above which has the POST form for the user to enter their username, then create the file reset-password-sms.php which contains the AWS client to send the SMS to the user

Create the following php file *reset_password_page.php* with the following code which has the POST form for the user to enter in their username.

- *reset_password_page.php*: <https://pastebin.com/Ey5gj4Lv>

Continue by creating another php file *reset-password-sms.php* which contains the AWS client to send the SMS to the user.

- *reset-password-sms.php*: <https://pastebin.com/jVs9hjSF>

Profile

Profile Page

Profile page code: <https://pastebin.com/7JXRBVH9>

Create the display-profile.php file with the code above in the root directory.

The code above will display all information of the session user if no GET method is available including their posts and the option to change profile picture and write a post. If a GET method is available then it will display the details of another user and the option to connect to the user.

Update personal profile

Update profile page code: <https://pastebin.com/31chFMNG>

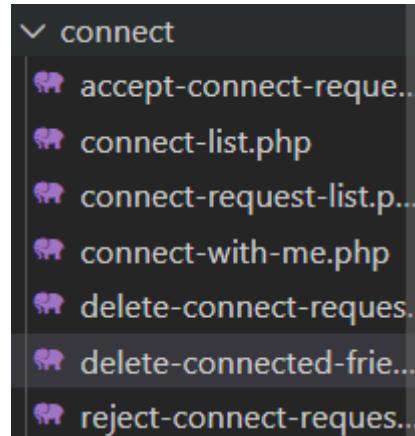
Create the update-profile.php file with the code above in the root directory. The code contains a POST form which will call the UpdatePersonalInfo() function in DynamodbFunctions class to update the user's details on DynamoDB.

Connect With Friends

Create a folder and name it 'connect'. Create the following php files inside the folder:

- *accept-connect-request.php*

- *connect-list.php*
- *connect-request-list.php*
- *connect-with-me.php*
- *delete-connect-request.php*
- *delete-connected-friend.php*
- *reject-connect-request.php*



Search for friends

Search user code: <https://pastebin.com/W8uGgvMT>

The code will use the SearchUser() function in DynamoDBFunctions to search for a list of users with the specified name.

Send connect request

Connect-with-me.php: <https://pastebin.com/bcCr7ZFB>

The 'Connect with me' button in profile page will direct to this php function to send a connect request to the target user

Undo connect request

delete-connect-request.php: <https://pastebin.com/ArStSus5>

The 'Undo connect request' button in profile page will direct to this php function to delete the connect request a user sent

Accept connect request

Connect request list page: <https://pastebin.com/EHrfi2m0>

Accept-connect-request.php: <https://pastebin.com/cWrssU4k>

The connect request list page will display all the connect requests a user has received, if the accept button is clicked the post method will call the accept-connect-request.php to add the other users as a friend stored in dynamoDB

Reject connect request

Reject-connect-request.php: <https://pastebin.com/KuZABvKD>

If the reject button is clicked in the connect request list page, the post method will call the reject-connect-request.php to remove the connect request in the dynamoDB table

View list of friends

connect-list.php: <https://pastebin.com/AKagt3nV>

This page will display the list of connected friends a user has

Delete a connected friend

Delete-connected-friend.php: <https://pastebin.com/FWTRReJxw>

If the Remove button is clicked in the Connect List page, the friend will be removed from the dynamoDB table

Posts

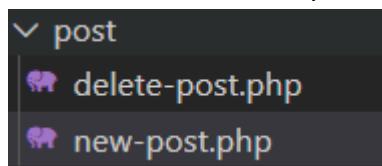
News feed

Code for newsfeed.php: <https://pastebin.com/Smgh2KbS>

Paste the code in the root directory of the project. The code will display the posts of all connected friends using multiple functions from DynamoDBFunctions class, S3Functions class and AwsTranslateFunctions class

Create a new post & delete post

Create a folder named 'post' and create 2 files inside as such:



new-post.php: <https://pastebin.com/FcyxbMRk>

This form allows the user to add some content and an image to the post as well as choosing the language

Delete-post.php: <https://pastebin.com/PjcFRfdx>

Each post made by the user logged in will have a 'Delete' button beside it, if clicked it will call the delete-post.php function to delete the post info from dynamoDB

Creating Tables On DynamoDB

FriendList table: Partition key is username and sort key is friendname

A screenshot of the AWS DynamoDB console. On the left, there is a sidebar with a 'Create table' button and a 'Delete table' button. Below these are buttons for 'Overview', 'Items', 'Metrics', 'Alarms', and 'Cap'. A search bar says 'Filter by table name' with a dropdown menu 'Choose a table ...' and an 'Actions' dropdown. A list of tables is shown with 'FriendList' selected. To the right, the 'Items' tab is selected. It shows a 'Scan' operation for the 'FriendList' table. The results table has columns 'username' and 'friendname'. There are five items listed: 'alice' with 'friendname' 'sean2', 'bob' with 'friendname' 'alice', 'bob' with 'friendname' 'sean2', 'carrotpop' with 'friendname' 'sean1', and 'carrotpop' with 'friendname' 'sean2'.

| username | friendname |
|-----------|------------|
| alice | sean2 |
| bob | alice |
| bob | sean2 |
| carrotpop | sean1 |
| carrotpop | sean2 |

FriendRequest table: Partition key is username and sort key is targetname

The screenshot shows the AWS Lambda console interface. On the left, a sidebar lists several tables: FriendList, FriendRequest (which is selected), Messages, Posts, and Users. The main area is titled "FriendRequest" and shows the "Items" tab selected. A search bar at the top says "Scan: [Table] FriendRequest: username, targetname". Below it, there's a "Scan" dropdown set to "Scan", a "Table" dropdown set to "[Table] FriendRequest: username, targetname", and a "Start search" button. The results table has columns for "username" and "targetname". Two items are listed: "admin" with "targetname" "bob", and "bob" with "targetname" "alice".

Messages table: Partition key is friendpair and sort key is timestamp

The screenshot shows the AWS Lambda console interface. On the left, a sidebar lists several tables: FriendList, FriendRequest, Messages (which is selected), Posts, and Users. The main area is titled "Messages" and shows the "Items" tab selected. A search bar at the top says "Scan: [Table] Messages: friendpair, timestamp". Below it, there's a "Scan" dropdown set to "Scan", a "Table" dropdown set to "[Table] Messages: friendpair, timestamp", and a "Start search" button. The results table has columns for "friendpair", "timestamp", "author", and "content". Five items are listed, all under the key "alice-sean2": timestamp 1611664020 (author sean2, content one), timestamp 1611664063 (author sean2, content two), timestamp 1611664071 (author sean2, content three), timestamp 1611664084 (author alice, content four), and timestamp 1611664089 (author alice, content five).

Posts table: Partition key is username and sort key is timestamp

Posts Close

Overview Items Metrics Alarms Capacity Indexes

Create item Actions

Scan: [Table] Posts: username, timestamp ^

Scan [Table] Posts: username, timestamp

+ Add filter

Start search

| | username | timestamp | content |
|--|-----------|------------|----------------------|
| | carrotpop | 1611557182 | ree |
| | carrotpop | 1611557327 | ree |
| | sean2 | 1611791758 | 我喜欢吃巧克力 |
| | vannieuwu | 1611547715 | dis is veri coool :D |

Users table: Partition key is username and sort key is user_type

Users Close

Overview Items Metrics Alarms

Create item Actions

Scan: [Table] Users: username, user_type ^

Scan [Table] Users: username, user_type

+ Add filter

Start search

| | username | user_type |
|--|-----------|-----------|
| | admin | User |
| | alice | User |
| | bob | User |
| | carrotpop | User |

1. S3 Bucket configurations

Create an S3 Bucket named 'imagesfbfile' with folder post and profile, if a different name is chosen the bucket name in the php functions needs to be changed as well. This bucket will store user profile images and post images

The screenshot shows the Amazon S3 console interface. At the top, it says "Amazon S3 > imagesfblite". Below that is the bucket name "imagesfblite" with a "Publicly accessible" badge. A navigation bar with tabs "Objects" (highlighted in orange), "Properties", "Permissions", "Metrics", "Management", and "Access Points" follows. Under the "Objects" tab, there's a section titled "Objects (2)". A toolbar below the table includes "Create" (disabled), "Delete", "Actions ▾", "Create folder", and "Upload". A search bar says "Find objects by prefix". The table lists two items:

| | Name | Type | Last modified |
|--------------------------|----------|--------|---------------|
| <input type="checkbox"/> | post/ | Folder | - |
| <input type="checkbox"/> | profile/ | Folder | - |

Change the Bucket policy to 'AllowPublicRead'

The screenshot shows the "Bucket policy" editor. At the top, it says "Bucket policy" and "The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)". There are "Edit" and "Delete" buttons. The main area contains the following JSON code:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "AWS": "*"
    },
    {
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::imagesfblite/*"
    },
    {
      "Sid": "2",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity E1EWXVCOHJJ1TG"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::imagesfblite/*"
    }
  ]
}
```

2. CloudFront configurations

Create a CloudFront distribution and set the origin to the s3 bucket you create in step 10, there will be a drop down menu to easily select it

CloudFront Distributions > E10NZXWAXMELK6

The screenshot shows the 'Origins' section of the CloudFront distribution configuration. It includes a table with columns: Origin Domain Name and Path, Origin ID, Origin Shield Region, Origin Type, and Origin Access Identity. There is one entry: 'imagesfblite.s3.amazonaws.com/post' with 'S3-Imagesfblite' as the Origin ID and 'S3 Origin' as the Origin Type.

| | Origin Domain Name and Path | Origin ID | Origin Shield Region | Origin Type | Origin Access Identity |
|--------------------------|------------------------------------|-----------------|----------------------|-------------|------------------------|
| <input type="checkbox"/> | imagesfblite.s3.amazonaws.com/post | S3-Imagesfblite | - | S3 Origin | - |

Get the Cloudfront 'Domain Name' and put it in the S3Functions Class in our php code

CloudFront Distributions > E10NZXWAXMELK6

The screenshot shows the 'General' tab of the CloudFront distribution configuration. It displays various settings such as Distribution ID (E10NZXWAXMELK6), ARN (arn:aws:cloudfront::251590602248:distribution/E10NZXWAXMELK6), Log Prefix (-), Delivery Method (Web), and more. The 'Domain Name' listed is d22qr0ael8s008.cloudfront.net.

| | |
|---------------------------------|--|
| Distribution ID | E10NZXWAXMELK6 |
| ARN | arn:aws:cloudfront::251590602248:distribution/E10NZXWAXMELK6 |
| Log Prefix | - |
| Delivery Method | Web |
| Cookie Logging | Off |
| Distribution Status | Deployed |
| Comment | - |
| Price Class | Use All Edge Locations (Best Performance) |
| AWS WAF Web ACL | - |
| State | Disabled |
| Alternate Domain Names (CNAMEs) | - |
| SSL Certificate | Default CloudFront Certificate (*.cloudfront.net) |
| Domain Name | d22qr0ael8s008.cloudfront.net |
| Custom SSL Client Support | - |
| Security Policy | TLSv1 |
| Supported HTTP Versions | HTTP/2, HTTP/1.1, HTTP/1.0 |
| IPv6 | Enabled |
| Default Root Object | - |
| Last Modified | 2021-01-28 20:56 UTC+11 |
| Log Bucket | - |

```
class S3Functions
{
    function __construct()
    {
        $this->s3Client = new S3Client([
            'region' => 'ap-southeast-2',
            'version' => 'latest'
        ]);

        // cloudfront
        $this->cloudFront = new CloudFrontClient([
            'region' => 'ap-southeast-2',
            'version' => 'latest',
        ]);

        // S3 BUCKET URLs
        $this->profileUrl = 'https://imagesfblite.s3-ap-southeast-2.amazonaws.com/profile/';
        $this->postUrl = 'https://imagesfblite.s3-ap-southeast-2.amazonaws.com/post/';

        // CloudFront URLs
        $this->cloudFrontUrl = 'https://d22qr0ael8s008.cloudfront.net/';
    }
}
```

3. Deploying to Elastic Beanstalk

Elastic Beanstalk

Platform: PHP
Platform branch: PHP 7.4 running on 64bit Amazon Linux 2
Platform version: 3.1.4 (Recommended)

Application code

- Sample application
- Existing version
- Upload your code**

Version label: testapplication-source

Source code origin: Maximum size 512 MB

Choose PHP as the platform for the Elastic beanstalk application and upload a zip folder containing all the code.

4. IAM Configurations

The elastic beanstalk application will create an IAM Role called `aws-elasticbeanstalk-ec2-role` which you need to attach `AmazonS3FullAccess` and `AmazonDynamoDBFullAccess` on it in order to access your DynamoDB and S3 Bucket

| Role name | Trusted entities | Last activity |
|--|--|---------------|
| <input type="checkbox"/> aws-elasticbeanstalk-ec2-role | AWS service: ec2 | None |
| <input type="checkbox"/> aws-elasticbeanstalk-service-role | AWS service: elasticbeanstalk | None |
| <input type="checkbox"/> AWSServiceRoleForAutoScaling | AWS service: autoscaling (Service-Linked role) | None |
| <input type="checkbox"/> AWSServiceRoleForElasticLoadBalancing | AWS service: elasticloadbalancing (Service-...) | None |
| <input type="checkbox"/> AWSServiceRoleForSupport | AWS service: support (Service-Linked role) | None |
| <input type="checkbox"/> AWSServiceRoleForTrustedAdvisor | AWS service: trustedadvisor (Service-Linked ...) | None |

Policy AmazonDynamoDBFullAccess has been attached for the aws-elasticbeanstalk-ec2-role.

| | | |
|---------------------------------|--|----------------------|
| Role ARN | arn:aws:iam::046119214116:role/aws-elasticbeanstalk-ec2-role | Edit |
| Role description | Edit | |
| Instance Profile ARNs | arn:aws:iam::046119214116:instance-profile/aws-elasticbeanstalk-ec2-role | Edit |
| Path | / | |
| Creation time | 2021-01-29 15:43 UTC+1100 | |
| Last activity | Not accessed in the tracking period | |
| Maximum session duration | 1 hour Edit | |

Permissions **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

▼ Permissions policies (5 policies applied)

Attach policies

| Policy name | Policy type |
|--|--------------------|
| ▶ AmazonS3FullAccess | AWS managed policy |
| ▶ AmazonDynamoDBFullAccess | AWS managed policy |
| ▶ AWSElasticBeanstalkWebTier | AWS managed policy |
| ▶ AWSElasticBeanstalkMulticontainerDocker | AWS managed policy |
| ▶ AWSElasticBeanstalkWorkerTier | AWS managed policy |

5. Refresh the Elastic Beanstalk instance

After a refresh the policies should be update and a working project is now properly hosted on elastic beanstalk

The screenshot shows the AWS Elastic Beanstalk console for environment V9-env. The left sidebar lists environments, applications, and configurations. The main area displays the V9-env details, including its URL (v9-env.eba-nhm3yvem.ap-southeast-2.elasticbeanstalk.com) and application name (v9). The Health section shows a green checkmark indicating 'Ok'. The Platform section shows 'PHP 7.4 running on 64bit Amazon Linux 2/3.1.4'. Below the console is a screenshot of a web browser displaying a 'Login Page' with a 'Sign In' form, which includes fields for 'Username' and 'Password' and a 'Login' button.

Messaging

Create a ‘messages’ folder and within the folder, create a ‘messaging.php’ file. Copy and paste the following pastebin code into the php file. This file will allow users to chat to each other.

- Messaging page code: <https://pastebin.com/yUmgIPnW>

Support page

This page allows users to send a support ticket to developers through AWS SES service. Begin by creating a 'support' folder and within the folder, create the following php files:

- 'send.php' and 'contact-form.php'

Paste each of the following pastebin codes into the corresponding files and replace your AWS SES username and password into the lines 27 and 30 respectively. These can be obtained from the AWS Dashboard.

- *send.php*: <https://pastebin.com/pqRNJ5qJ>
- *contact-form.php*: <https://pastebin.com/1qZz1m5R>

Logout

This page allows users to terminate their session and redirects them to the login page. Create a file php file in the root folder and name it 'logout.php'. Copy and paste the following pastebin code into the php file.

- *logout.php*: <https://pastebin.com/iKSfsFJZ>

h. AWS Services Used

- **Criteria Two**

- 1. **AWS Cloudwatch (Approved by the lecturer):**

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health^[1].

CloudWatch Log:

The screenshot shows an email from AWS Notifications (no-reply@sns.amazonaws.com) to Clinton Pham on 28/01/2021 at 7:54 PM. The subject is "ALARM: "Budget" in US East (N. Virginia)". The email body contains the following text:

You are receiving this email because your estimated charges accrued for this billing period are currently \$ 2.02 USD as of Thursday 28 January, 2021 08:54:20 UTC. The actual charges you will be billed in this statement period may differ from the charges shown on this notification. For more information, view your estimated bill at: [View this alarm in the AWS Management Console: <https://us01.usfelinks.protection.outlook.com/?url=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fcloudwatch%2Fhome%3Fregion%3Dus-east-1%2523%2Dalarms%26alarmId%3DBudget&data=04%7C01%7C3605044%40student.rmit.edu.au%7Cfafb091256894d83115c08de417b4d4bd24b513168%7C0%7C63742742067065309%7CUnknown%7CTWFpbGZhb3phbGxeyIWjoiMC4wIjAwMDAiCQjoiV2luMzliCJBtI6lk1haWwliCXBvI6Mn0%3D%7C1000&sdata=E%2BGriQ8%2BmNUT4tWhlRzYmoOAvkZgJU4WPVjrhw%3D&reserved=0>

Alarm Details:

- Name: Budget
- Description: INSUFFICIENT_DATA -> ALARM
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints \[2.02 \(28/01/21 02:54:00\)\] was greater than the threshold \(1.0\) \(minimum 1 datapoint for OK -> ALARM transition\).
- Timestamp: Thursday 28 January, 2021 08:54:20 UTC
- AWS Account: 103237653301
- Alarm Arn: arn:aws:cloudwatch:us-east-1:103237653301:alarm:Budget

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 1.00 for 21600 seconds.

Monitored Metric:
- MetricNamespace: AWS/Billing
- MetricName: EstimatedCharges
- Dimensions: \[Currency = USD\]
- Period: 21600 seconds
- Statistic: Maximum](https://us01.usfelinks.protection.outlook.com/?url=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fbilling%2Fhome%23%2Fbill%3Fyear%3D2021%26month%3D1&data=04%7C01%7C3605044%40student.rmit.edu.au%7Cfafb091256894d83115c08de417b4d4bd24b513168%7C0%7C63742742067065309%7CUnknown%7CTWFpbGZhb3phbGxeyIWjoiMC4wIjAwMDAiCQjoiV2luMzliCJBtI6lk1haWwliCXBvI6Mn0%3D%7C1000&sdata=4CPIO4Au%7CTVsSeJcdwRjg4Bd7sW62WpoPw%3D&reserved=0</p><p>More details about this alarm are provided below:</p><p>Amazon CloudWatch Alarm)

Approved Email:

The screenshot shows two emails exchanged between Andrian Radic and Clinton Pham.

Email 1 (From: Andrian Radic, To: Clinton Pham; Sean Tan)

- Subject: [REDACTED]
- Date: Thu 28/01/2021 8:42 AM
- Content:
 - Hi,
 - Yes that's fine.
 - Andrian
 - ...
- Buttons: Reply | Reply all | Forward

Email 2 (From: Clinton Pham, To: Andrian Radic; Sean Tan)

- Subject: [REDACTED]
- Date: Wed 27/01/2021 4:10 PM
- Content:
 - Hi Andrian,
 - Hope you are doing well.
 - To get full marks in '**Appropriate utilization of cloud tools/technologies/services in your project (27 marks)**' we need to utilise **8 cloud services**.
 - **6** type of cloud services from criteria 2 - **Criteria 2** states: The use of Google/AWS cloud services.
 - My partner and I have been working hard on the assignment and we are wondering whether **AWS Cloud9** would be considered as one service for Criteria 2 in the assignment specification. AWS Cloud 9 is a development environment that multiple users can participate in. We have been using this because assignment 2 is a group assignment and cloud9 has many features useful to us such as **pair programming** feature which allow us work on the same project in a single environment. This allow us to see live changes in the code without being next to each other in person.
 - Is it safe to assume that as long as we make use of any services in the AWS (other than database/host from criteria 3 and 4) it would be valid for criteria 2?
 - We have also integrated other services such as SNS and SES but have also made use of **CloudWatch** to ensure we don't go beyond our budget. Even though we're not implementing CloudWatch and Cloud9 directly into our php application (but have used it on the aws platform) would it be okay to add them to the criteria 2 list?
- Buttons: Reply | Reply all | Forward

2. AWS Amazon Transcribe

Amazon Transcribe makes it easy for developers to add speech to text capabilities to their applications^[2].

The screenshot shows the AWS Amazon Transcribe interface.

Navigation bar: News Feed, Messages, My Profile, Update Profile, Connected Friends, Connect Requests, Fun Implementations!, Logout, Find Friends

Main content area:

- Text: Try out our awesome audio to text feature!
- Input field: Choose file No file chosen
- Text area: Hey! No attention to them. My speech here is my Christmas speech. Thank you all. And merry Christmas.
- Button: Transcribe into text!

3. AWS Simple Email Service (SES)

Amazon Simple Email Service (SES) is a cost-effective, flexible, and scalable email service that enables developers to send mail from within any application^[3]

4. AWS Simple Notification Service (SNS)

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication^[4]



5. AWS Cloudfront

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment^[5].

6. AWS EC2

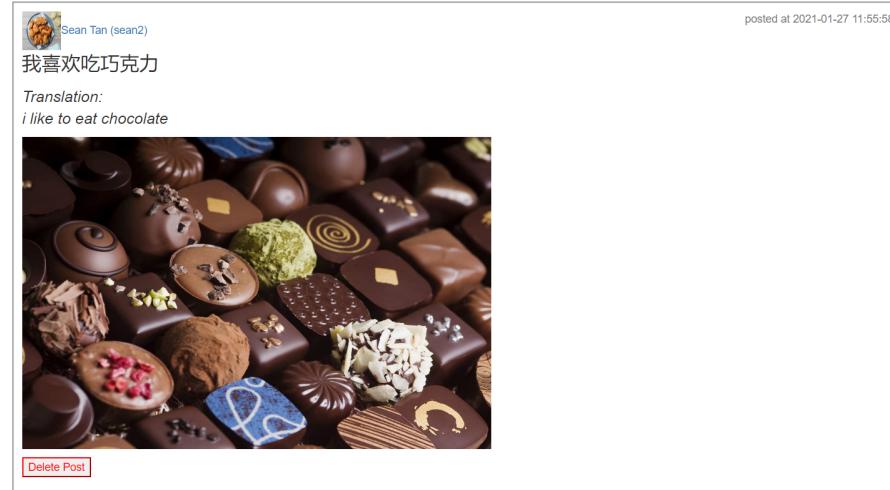
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction^[6].

7. AWS Translate

Amazon Translate is a neural machine translation service that delivers fast, high-quality, and affordable language translation. Neural machine translation is a form of language translation automation that uses deep

learning models to deliver more accurate and more natural sounding translation than traditional statistical and rule-based translation algorithms^[7].

My Posts



8. AWS Cloud9 (Approved by the lecturer)

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal^[8].

The screenshot shows the AWS Cloud9 IDE interface. On the left is a file browser with a tree view of files and folders. The main area has a code editor with the following PHP code:

```
<?php
$curl = "https://www.coinspot.com.au/pubapi/latest";
$content = file_get_contents($curl);
$json = json_decode($content, true);

// Check if the form is submitted
if (!isset($_GET['submit'])) {
    // retrieve the form data by using the element's name attributes value as key
    $cryptoOne = $_GET['crypto-one'];
    $cryptoTwo = $_GET['crypto-two']; // display the results
    $cryptoThree = $_GET['crypto-three'];

    echo '<h3>Form GET Method</h3>';
    echo 'Price for ' . $cryptoOne . ' is: ' . $json['prices'][$cryptoOne]["last"] . '<br>';
    echo 'Price for ' . $cryptoTwo . ' is: ' . $json['prices'][$cryptoTwo]["last"] . '<br>';
    echo 'Price for ' . $cryptoThree . ' is: ' . $json['prices'][$cryptoThree]["last"] . '<br>';
}
?>
```

Below the code editor is a terminal window showing the command `php -f ip-172-31-17-56.ec` and the output "front_end/display-user.php". At the bottom, there are tabs for "Run", "Command: front_end/display-user.php", and "Runner: PHP (built-in web server) CWD".

- **Criteria Three**

1. **S3 Bucket**

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance^[9].

2. **DynamoDB**

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications^[10].

- **Criteria Four**

1. **AWS Elastic Beanstalk**

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS^[11].

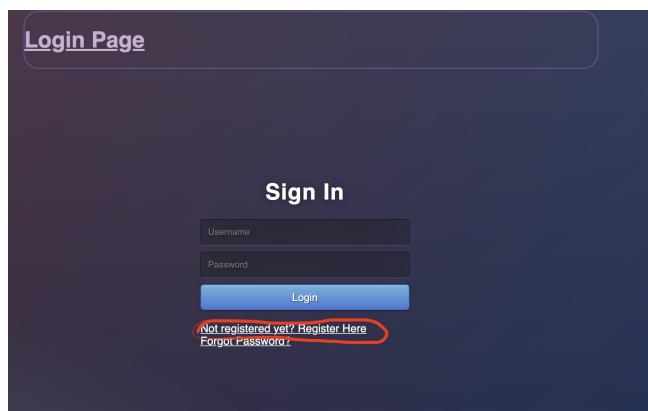
- i. A small user manual

Begin by heading over to the following link to access the webpage:

<http://v9-env.eba-nhm3yvem.ap-southeast-2.elasticbeanstalk.com/loginpage.php>

Registering for an account

In order to access the webpage, you will need an account. Proceed by clicking on 'Not registered yet? Register here' link to register your account. Fill in the required information.



After you have registered an account, use your credentials to sign into the page.

Update Profile

Click on the '*Update Profile*' link in the navigation bar to update your profile. Fill out the required information, this will be displayed in your profile.

Personal Info

Full Name *
Password *
Email *
Phone *
About Me *

Clinton Pham
.....
Clintonjob@hotmail.co.i
+61412645235
cloud computing

Favourite Cryptocurrencies

Crypto One: Bitcoin
Crypto Two: Ethereum
Crypto Three: Litecoin

Favourite Websites

Website1
Website2
Website3

google
facebook
youtube

Update Profile

Send An Email To The Support Team

Click on the '*Support Center*' in the navigation bar to be directed to the support page. In this page, you will be able to send an email over to the support department.

Support Center

For any enquires and issues please fill out the form below to be in contact with our support team.

Name *

Please enter your firstname *

Email Address *

Please enter your email address *

Subject/Issue

Subject goes here *

Message *

Please enter your message

Send message

Connect With Others or Find Friends

Click on the '*Find Friends*' button in the navigation bar to see a list of registered users in the database. From here, you will be able to click on the users profile and be directed to their profile which will allow you to connect to them.

Find Friends

Search Here...

Search

Bob (bob)

Sean Tan 2 (sean)

vannie (vannieuwu)

Clinton Pham (iClinton)

Alice2 (alice)

Go Back to Profile

After clicking on the users profile, click on '*Connect with me*' to send a connect request to the user. The user will be able to approve or reject your request.

News Feed Messages My Profile Update Profile Connected Friends Connect Requests

Support Center Logout

Find Friends

Welcome To Bob's Profile Page.

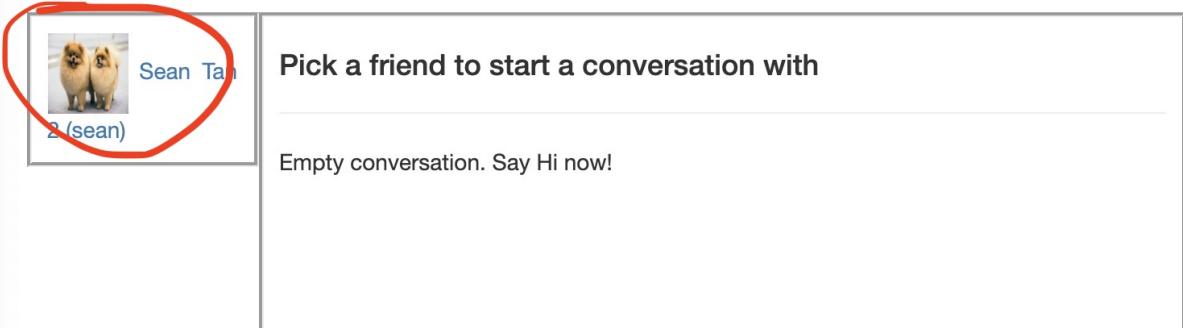
Connect with me

Bob (bob)

Send Messages

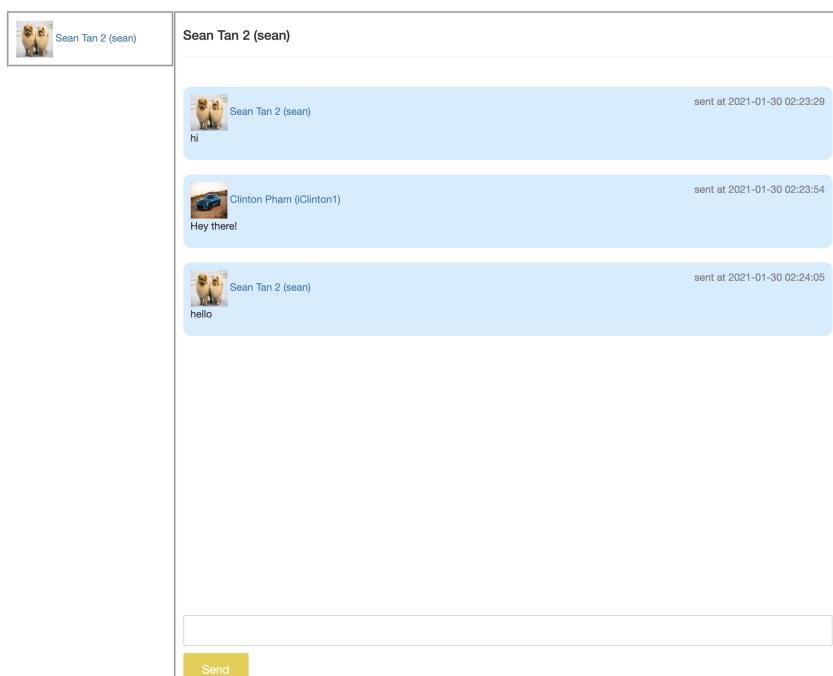
In order to send messages to your friends or family, you will both need to be connected to each other. Proceed by clicking on the '*Messages*' link in the navigation bar. You will see your connected friends on the left hand side with their profile image and name. To send them a message, click on the name.

Messages



A screenshot of a messaging interface. On the left, there is a card for a friend named "Sean Tan". The card features a small thumbnail image of two dogs, the name "Sean Tan" in blue text, and the handle "2 (sean)" below it. A red circle highlights this entire card. To the right of the card, the text "Pick a friend to start a conversation with" is displayed. Below this, the message "Empty conversation. Say Hi now!" is shown.

After clicking on the name, you will be able to communicate with your friend.



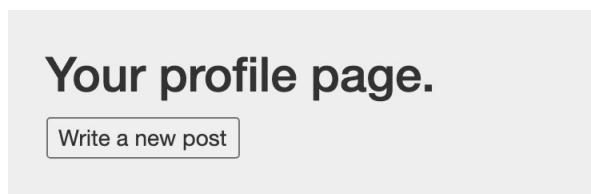
A screenshot of a messaging conversation. The left side shows a list of messages:

- Sean Tan 2 (sean) sent a message at 2021-01-30 02:23:29: "hi"
- Clinton Pham (@Clinton1) responded at 2021-01-30 02:23:54: "Hey there!"
- Sean Tan 2 (sean) responded at 2021-01-30 02:24:05: "hello"

At the bottom, there is a text input field and a yellow "Send" button.

Create a Post on your Profile Page

Click on 'My Profile' page to be directed to your profile page where you will be able to create a new post. Click on 'Write a new post' to create a new post.



A screenshot of a profile page. At the top, the text "Your profile page." is displayed in large, bold letters. Below this, there is a button labeled "Write a new post".

To upload an image to the post, click on ‘choose file’ and proceed to navigate to the file. Once a file has been selected, click on the upload button to upload the file to the database where it will be stored. Write a status in the text area. Click on post to upload the new post to your profile.

Write a new post

Write something...

Upload an image

No file chosen

Language

Translate Chinese to English

This feature translates non-english text to english text. Create a new post and in the text area, put in chinese text such as ‘你好， 我叫克林顿’. This translates to ‘Hello, my name is Clinton’. Select Chinese (Simplified) as the language and click on the post button. You will see your translated post on your wall.

My Posts



Clinton Pham (iClinton1)

posted at 2021-01-30 06:23:38

你好， 我叫克林顿

Translation:

Hello, my name is Clinton

j. References

1. <https://aws.amazon.com/cloudwatch/>
2. <https://aws.amazon.com/transcribe/>
3. <https://aws.amazon.com/ses/>
4. <https://aws.amazon.com/sns>
5. <https://aws.amazon.com/cloudfront>
6. <https://aws.amazon.com/ec2>
7. <https://aws.amazon.com/translate>
8. <https://aws.amazon.com/cloud9>
9. <https://aws.amazon.com/s3/>
10. <https://aws.amazon.com/dynamodb/>
11. <https://aws.amazon.com/elasticbeanstalk/>

k. Video (optional)

Page Demo: <https://www.youtube.com/watch?v=ICV4uAcVZyw&feature=youtu.be>

Note: Not all features we implemented were shown in the demonstration video.