



Technical Documentation

Cotiss Anonymous Feedback Web Application

LAST UPDATED: 15TH DECEMBER 2023

WRITTEN BY GAVIN LIM

Table of Contents

Contents

Table of Contents.....	2
Introduction	4
Product and Project Overview	4
Key Files	5
Index.php	5
Feedback.php.....	6
Step by Step Implementation Guide – Level One.....	7
Virtual Private Cloud	7
Creating a Virtual Private Cloud (VPC)	7
Creating your Subnets for your VPC	8
Create an Internet Gateway (IGW)	9
Editing the Route Table of the VPC.....	10
Creating a DynamoDB Table for Storing Feedback.....	11
Create the Table.....	11
Adding an Item.....	12
Setting up your EC2 Instance	13
Creating your EC2 Instance	13
Creating an IAM Role for your EC2 Instance.....	15
Add your Role to your EC2 Instance	17
Installing Software for your EC2 Instance.....	18
Upload your PHP Files into your EC2 Instance.....	19
Creating an AMI	20
Creating a Launch Template	20
Create your Auto Scaling Group and Load Balancer	22
Creating an Endpoint for your DynamoDB	25
Check your Work.....	26
Step by Step Implementation Guide – Level Two.....	27
Creating a .zip for your Elastic Beanstalk.....	27
Creating an Elastic Beanstalk Environment and Application	27

Adding our Policy to the EC2 Elastic Beanstalk Role.....	29
Have an S3 Bucket serve your Static Data	29
Create an S3 Bucket.	29
Upload your Static Content – Logo Image	29
Registering a Domain for Route 53.....	30
Creating a Hosted Zone	30
Creating a Hosted Zone.....	30
Requesting a Certificate.....	30
Creating CloudFront Distributions	31
Create a CloudFront Distribution for your Elastic Beanstalk Environment	31
Create Distribution for your Static Content Bucket.....	33
Adding your CloudFront Distributions as Records.....	34
Check your work	34
Cost Analysis	35
Level One	35
Level Two	35

Introduction

This technical documentation provides an overview and guide to implementing the Amazon Web Service (AWS) Solution created for Cotiss. The solution contains two levels, level one being the base implementation and level two being the slightly more advanced implementation. To complete level two, level one is required up until This documentation is provided alongside the [demo video](#) which will give a brief walkthrough of the solutions. This documentation assumes that you are familiar AWS and can navigate the AWS Console and has a general understanding of AWS and its services, as well as decent understanding of programming. This documentation also assumes that you have a decent understanding of the Linux Command Line and other tools such as SSH clients.

Product and Project Overview

The AWS solution was created for the purpose of collecting anonymous feedback from employees of Cotiss. Cotiss leadership wanted a simple website allowing this function to better collect feedback within the company. The website is to contain two key sections. Firstly, a random piece of previously submitted feedback is to be displayed. The incentive behind this is to encourage more honest feedback. Secondly, there is to be a feedback form which the user can submit to provide their own feedback. This feedback gets stored in a DynamoDB with no trace to the user who submitted it.

Important Note:

This product was built using only the free tier services provided by AWS.

Key Files

Index.php

This is the page that visitors of our webpage will see. You can find the full file [here](#).

If you do not plan on completing level two, you can find the level one [files](#) here.

The key features that are contained in this file are:

- The front-end of the website – All the content of the website will be in this file.
- Creating a DynamoDBClient using the factory method in the AWS SDK. This is important for the reading and displaying a random piece of feedback in the database. The parts highlighted green will vary depending on your region and the name of your DynamoDB Table that you will create later.

```
require 'vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

try {
    $client = DynamoDbClient::factory(array(
        'region' => 'ap-southeast-2',
        'version' => 'latest'
    ));

    $tableName = 'CotissFeedbackTable';

    $response = $client->scan(array(
        'TableName' => $tableName
    ));
```

- Reading a random piece of feedback and displaying it.
- Sending the feedback to our other PHP file which will submit the feedback into our DynamoDB Table.

Feedback.php

This file is responsible for taking the feedback the user has provided and inputting it into our DynamoDB Table and returning the user back to the landing page. This file is purely back-end. You can find the full file [here](#).

The key features that are contained in this file are:

- Taking the feedback provided by the user from index.php and adding it to the DynamoDB Table. The parts highlighted green will vary depending on your region and the name of your DynamoDB Table that you will create later.

```
require 'vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

try {
    $client = DynamoDbClient::factory(array(
        'region' => 'ap-southeast-2',
        'version' => 'latest'
    ));

    $tableName = 'CotissFeedbackTable';

    $response = $client->putItem(array(
        'Item' => array(
            'id' => array('N' => getUniqueId($client, $tableName)),
            'feedback' => array('S' => $feedback),
            'rating' => array('N' => $rating)
        ),
        'TableName' => $tableName
    ));
```

- Generating a unique ID for the DynamoDB Table entry
 - This is currently done using an incrementation method. This is not ideal but is functional for the purposes that we have.
- Returning the user to the landing page (index.php)

Step by Step Implementation Guide – Level One

Level One consists of using EC2 instances to host the website using Apache. The use of a load balancer and auto scaling group allowed for better availability and elasticity for this solution. This section will share the steps required to achieve this outcome.

Virtual Private Cloud

Creating a Virtual Private Cloud (VPC)

1. Make sure to select your desired region in the top right. This will be the region which your VPC will be deployed. Sydney is the closest region available region for Cotiss as of the date of this documentation and will be the region this documentation follows.
2. Follow the settings as outlined below:
 - Select VPC only
 - Name your VPC – ours is called CotissFeedbackVPC.
 - Keep the default option of IPv4 CIDR manual input.
 - Insert an IPv4 CIDR. A /16 bit mask is appropriate as this will need to be divided into smaller subnets for the different availability zones. The IPv4 CIDR used in our solution is 168.16.0.0/16 and we recommend following this for ease of following the document.
 - Everything else can be kept as default. Click Create VPC.

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☒ VPC only

☐ VPC and more

Name tag - optional

Creates a tag with a key of 'Name' and a value that you specify.

VPC NAME

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input

☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

168.16.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

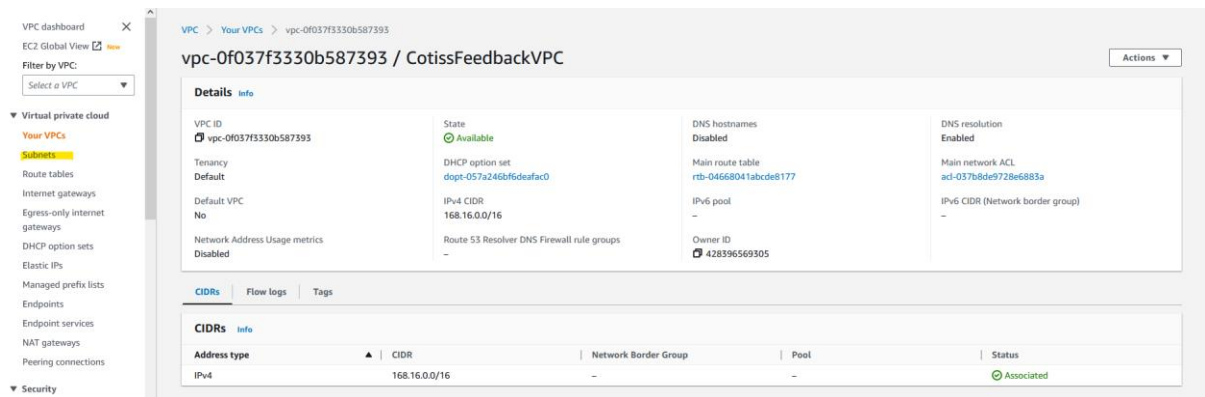
☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

Creating your Subnets for your VPC

1. Navigate to Subnets. This is located on the left-hand menu. This documentation will refer to this left-hand menu occasionally.



2. Select Create Subnet in the top right corner of this screen.
3. Under VPC ID, select the VPC you created previously.
4. Create a subnet with the following settings:
 - Insert a subnet name.
 - Select one of your Availability Zones. If you are in Sydney – there should be 3.
 - Enter 168.16.0.0/20 as your IPv4 CIDR block.

VPC

VPC ID
Create subnets in this VPC.
vpc-0f037f3330b587393 (CotissFeedbackVPC)

Associated VPC CIDRs
IPv4 CIDRs
168.16.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
CotissFeedbackSubnet01
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Sydney) / ap-southeast-2a

IPv4 CIDR block [Info](#)
168.16.0.0/20

Tags - optional

Key	Value - optional	
Name	CotissFeedbackSubnet01	Remove

Add new tag

You can add 49 more tags.

5. Repeat the bullet points in Step 4 until you have 3 different Subnets with the following changes for each of the new subnets:
 - Use a different Availability Zone for each subnet.
 - Set the IPv4 CIDR block to 168.16.16.0/20 and 168.16.32.0/20 for your 2nd and 3rd subnet respectively.
6. Ensure that you have 3 different subnets in 3 different availability zones.

Subnets (3) Info										
<input type="text" value="Filter subnets"/> Actions Create subnet										
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6...	Aval...	Availability Zone	Availability Zone ID	
<input type="checkbox"/>	CotissFeedbackSubnet02	subnet-07b8e7e0fc0ba35ee	Available	vpc-0f037f3330b587393 Cot...	168.16.16.0/20	–	4090	ap-southeast-2b	apse2-az3	
<input type="checkbox"/>	CotissFeedbackSubnet03	subnet-05d289c0fa48e067	Available	vpc-0f037f3330b587393 Cot...	168.16.32.0/20	–	4090	ap-southeast-2c	apse2-az2	
<input type="checkbox"/>	CotissFeedbackSubnet01	subnet-0c86306d413bd2350	Available	vpc-0f037f3330b587393 Cot...	168.16.0.0/20	–	4090	ap-southeast-2a	apse2-az1	

Create an Internet Gateway (IGW)

1. Navigate to Internet Gateway on the left-hand menu.

2. Create a new Internet Gateway - remember its name as it will be in the following section.

3. Select Actions and then Attach to VPC
4. Select the VPC that you created earlier and click Attach Internet Gateway

Editing the Route Table of the VPC

1. Navigate back to the VPC located in the left-hand menu and select its main route table.

Details
[Info](#)

VPC ID vpc-0f037f3330b587393	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-057a246bf6deaaf0	Main route table rtb-04668041abcde8177	Main network ACL acl-037b8de9728e6883a
Default VPC No	IPv4 CIDR 168.16.0.0/16	IPv6 pool -	IPv6 CIDR -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 428396569305	

[CIDRs](#)
[Flow logs](#)
[Tags](#)

CIDRs
[Info](#)

Address type	CIDR	Pool	Status
IPv4	168.16.0.0/16	-	Associated

2. Edit the routes in the route table. Add the following route. Set the Destination as 0.0.0.0/0. This sets the incoming destination as anywhere.
3. Set the target as the IGW that you created before. You can find the IGW you created by identifying its name which is shown in brackets next to the long IGW identifier.
4. Ensure you save your changes.

Q 0.0.0.0/0	X	Q igw-03b797460b273046b	X	Active	No	Remove
-------------	---	-------------------------	---	--------	----	--------

Creating a DynamoDB Table for Storing Feedback

Create the Table

1. Navigate to your DynamoDB Dashboard
2. Create a new DynamoDB Table with the following settings:
 - The name of your table is up to your choosing – ensure you remember the name of the table for later use.
 - Set your partition key to a number. In our solution, we have named it id.
 - A sort key is optional, one was not used in our solution. A potential sort key could be the rating of the feedback which we will add later.
 - The remaining settings can remain unchanged.

Table details [Info](#)
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
 ▼
1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
 ▼
1 to 255 characters and case sensitive.

Table settings

☒ **Default settings**
The fastest way to create your table. You can modify these settings now or after your table has been created.

☐ **Customize settings**
Use these advanced features to make DynamoDB work better for your needs.

3. Wait for your DynamoDB Table to finish creating.

Adding an Item

This step creates a guideline/template for all further item additions to our DynamoDB Table.

1. Once your DynamoDB Table has finished creating, click and view the table.
2. Create a new item for your DynamoDB Table – follow the guidelines below:
 - Click on Actions in the top right and then select Create Item.
 - The id can remain as 0.
 - Add a new attribute of type number. This will be named “rating”.
 - Set the rating to a number between 1-3. This represents negative, neutral or positive.
 - Add a new attribute of type string. This will be named “feedback”.
 - Set the feedback to anything of your choosing – do ensure that this field is filled out.
 - Create the Item

Attributes			Add new attribute ▼
Attribute name	Value	Type	
id - Partition key	0	Number	
rating	2	Number	Remove
feedback	Awesome!	String	Remove

3. Ensure that you can see the item in your DynamoDB Table.

Setting up your EC2 Instance

Creating your EC2 Instance

If you plan on implementing Level Two without doing Level One – you do not need to complete this section and all subsequent Level One sections except “Creating an Endpoint for DynamoDB” which you will need to complete.

1. Navigate to your EC2 Dashboard.
2. Launch a new EC2 Instance. Follow the settings below:
 - Add a name for your new instance. This can be anything.
 - Keep Amazon Linux as your OS Image and the architecture as 64-bit (x86)

The screenshot shows the AWS Management Console interface for launching an EC2 instance. The 'Name and tags' section has a text input field with 'CotissFeedbackInstance' and a link to 'Add additional tags'. The 'Application and OS Images (Amazon Machine Image)' section is expanded, showing a search bar and tabs for 'Recents', 'My AMIs', and 'Quick Start'. Under 'Quick Start', there are tiles for 'Amazon Linux', 'macOS', 'Ubuntu', 'Windows', and 'Red Hat'. The 'Amazon Linux' tile is selected, showing details for 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type'. The architecture is set to '64-bit (x86)' and the AMI ID is 'ami-051a81c2bd3e755db'. A 'Verified provider' badge is visible.

- Instance Type can remain as **t2.micro**.
- Create a new Key Pair (unless you have one which you would like to use already).
 - Keeping RSA as your pair type is appropriate.
 - Depending on your preferred SSH method, using either .pem or .ppk is appropriate.
Note: PuTTY/.ppk is for Windows only.

The screenshot shows the 'Create key pair' dialog box. It explains that key pairs allow secure connection to the instance and prompts the user to enter a name. The 'Key pair name' field contains 'CotissFeedbackKeyPair'. Under 'Key pair type', 'RSA' is selected. Under 'Private key file format', '.pem' is selected. At the bottom, there are 'Cancel' and 'Create key pair' buttons.

3. Press Edit in the top right of Network Settings and apply the following changes.
 - Under VPC, select the VPC that you created in the previous steps.
 - Under Subnet, select any of the subnets that you created previously.
 - Enable Auto-assign public IP.
 - Create a new Security Group with SSH and HTTP. Give this group a name and description.
 - Allow SSH from your IP only for best security. If working in a team, you will need to identify an appropriate rule to allow all those working on it to SSH into the instance or just have one person access it. Note that if your ISP may change your public IP address so you may find that you can no longer SSH into your instance. You will need to change your rule if that is the case.
 - Allow HTTP from **anywhere (0.0.0.0/0)** so people can access the webpage which you will eventually launch.

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - *required*
CotissSG
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./@#%&()*+&#'\$*

Description - *required* [Info](#)
SG for Cotiss

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, [redacted]) [Remove](#)

Type	Protocol	Port range	Source type	Name	Description - optional
ssh	TCP	22	My IP	[redacted]	e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0) [Remove](#)

Type	Protocol	Port range	Source type	Source	Description - optional
HTTP	TCP	80	Anywhere	0.0.0.0/0	e.g. SSH for admin desktop

4. Launch your instance.

Creating an IAM Role for your EC2 Instance

1. Navigate to your IAM Dashboard.
2. Go to roles on your left-hand menu and select Create role.
3. Select AWS service and EC2.

Select trusted entity [Info](#)

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- ☒ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☐ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

Cancel

Next

4. Select Create Policy in the top right and then select JSON.
5. Insert the following – replace XXXXXXXXXXXX with your Account ID:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dynamodb:BatchGetItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:PutItem",
        "dynamodb:PartiQLSelect",
        "dynamodb:GetShardIterator",
        "dynamodb:GetItem",
        "dynamodb:PartiQLInsert",
        "dynamodb:Scan",
        "dynamodb:Query",
        "dynamodb:GetRecords"
      ],
      "Resource": [
        "arn:aws:dynamodb:*:XXXXXXXXXXXX:table/CotissFeedbackTable",
        "arn:aws:dynamodb:*:XXXXXXXXXXXX:table/CotissFeedbackTable/stream/*"
      ]
    }
  ]
}
```

6. Select Next: Tags.
7. Select Next: Review.
8. Give the Policy a name and description.
9. Create Policy.
10. Go back to the original tab.
11. Refresh the page and locate your new policy.
12. Select the checkbox next to your new policy and hit Next.
13. Give this role a name and description.
14. Scroll down and ensure that the trusted entities is correct (as below) and your policy name is displayed under permissions.

Step 1: Select trusted entities

Edit

```
1- [{
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": {
10-        "Service": [
11-          "ec2.amazonaws.com"
12-        ]
13-      }
14-    }
15-  ]
16-}]
```

Step 2: Add permissions

Edit

Permissions policy summary		
Policy name ↗	Type	Attached as
CotissFeedbackReadWriteDynamoDB	Customer managed	Permissions policy

15. Create role.


Add your Role to your EC2 Instance

If you plan on implementing Level Two without doing Level One – you do not need to complete this section and all following sections except “Creating an Endpoint for y

1. Navigate back to your EC2 Dashboard
2. View your instances and select the checkbox next to your instance.
3. Select Actions then Security then Modify IAM Role.
4. Add your newly created role to your instance and select Update IAM role.

Modify IAM role [Info](#)
Attach an IAM role to your instance.



Instance ID

 **i-06fc679fce45b49ae** (CotissHonestFeedback)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

EC2CotissFeedbackEditor ▼

 [Create new IAM role](#) 

Cancel

Update IAM role

Installing Software for your EC2 Instance

1. SSH into your EC2 instance using your preferred method using the key pair you created previously. If you are unsure on how to SSH, the following documentation may be useful.
 - [Using PuTTY for Windows](#) (.ppk key file)
 - [Using OpenSSH for Windows/Linux/MacOS](#) (.pem key file)
2. Update all your packages with the following command:
 - **sudo yum update -y**
3. Install Apache and PHP with the following command.
 - **sudo yum install httpd php -y**
4. Check that both Apache and PHP have been installed with the following commands.
 - **httpd -v**
 - **php --version**

```
[ec2-user@ip-168-16-31-120 ~]$ httpd -v
Server version: Apache/2.4.54 ()
Server built:   Jun 30 2022 11:02:23
[ec2-user@ip-168-16-31-120 ~]$ php --version
PHP 5.4.16 (cli) (built: Oct 31 2019 18:34:05)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
[ec2-user@ip-168-16-31-120 ~]$
```

5. Start your Apache server with the following command
 - **sudo systemctl start httpd**
6. Ensure that Apache launches on start-up with the following command
 - **sudo systemctl enable httpd**
7. Visit your public IP address and make sure it is HTTP. You should see a Test Page.

Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.


If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](#), you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

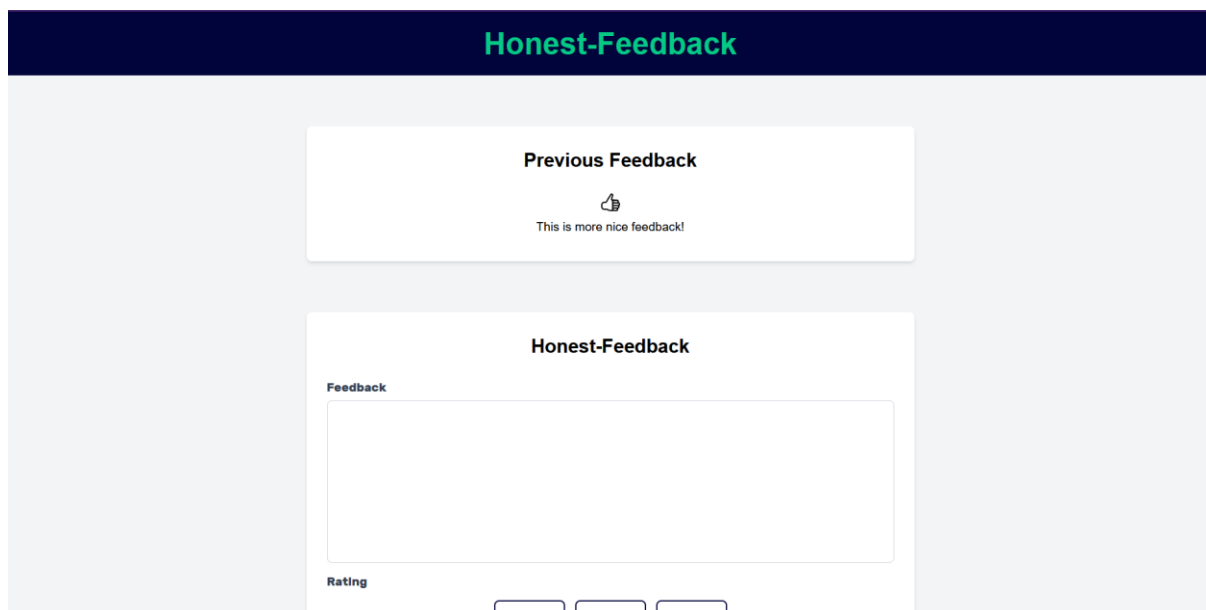
You are free to use the image below on web sites powered by the Apache HTTP Server:



8. Going back to your SSH instance, enter the following command.
 - `cd /var/www/html`
9. Run the following commands to install Composer and the AWS SDK for PHP.
 - `sudo php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"`
 - `sudo php composer-setup.php`
 - `sudo php -r "unlink('composer-setup.php');"`
 - `sudo php -d memory_limit=-1 composer.phar require aws/aws-sdk-php`

Upload your PHP Files into your EC2 Instance

1. Transfer the [Level One](#) PHP files and [CSS folder](#) into your EC2 instance. The files should then be moved into the `/var/www/html` folder. The following documentation may be useful.
 - [For Windows and MacOS](#)
 - [For Linux](#)
2. In your SSH client and in the `/var/www/html` directory run the `ls` command to check that your files are present in the folder.
3. Visit your website again. It should display the website correctly now.



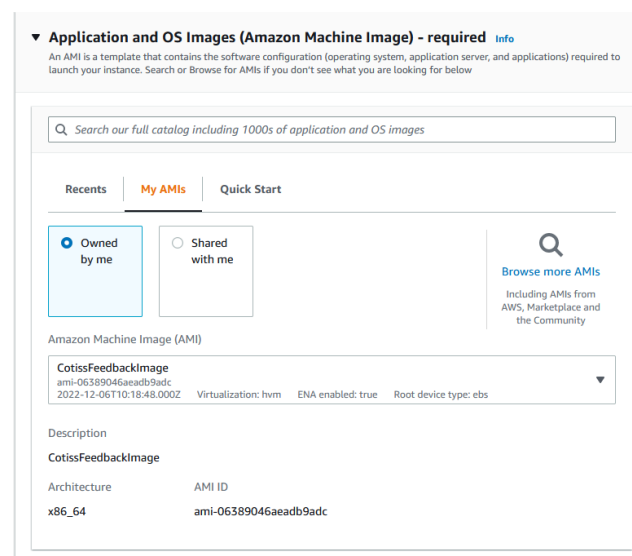
Creating an AMI

We want to save this instance state for a launch template which we will use for our Auto Scaling later.

1. Navigate back to your EC2 dashboard.
2. View your instances by clicking Instances in the left-hand menu.
3. Tick the checkbox next to your instance.
4. Select Actions in the top right, then Images and Templates then Create Image.
5. Give your image a name and then Create Image.

Creating a Launch Template

1. Navigate to Launch Templates on the left-hand menu.
2. Select Create Launch Template
3. Give your Launch Template a name.
4. Under Auto Scaling guidance, tick the checkbox as this will ensure that you don't miss any important parts.
5. Under Application and OS Images, click on My AMIs and find the Image that you just created.



6. Select your instance type as **t2.micro**.

- Under network settings, ignore Subnet and select the security group you created before.

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template ▼ [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group ☐ Create security group

Security groups [Info](#)

Select security groups ▼

CotissFeedbackSG sg-0c26b67f178138efb ✕
VPC: vpc-0f037f3330b587393 [Compare security group rules](#)

► Advanced network configuration

- Scroll down and show the Advanced details.
- In IAM instance profile, choose the role that you created earlier.

▼ Advanced details [Info](#)

Purchasing option [Info](#)

☐ Request Spot Instances

If Spot is selected you will not be able to create an Auto Scaling group that spans across multiple pricing options and instance types

IAM instance profile [Info](#)

EC2CotissFeedbackEditor
arn:aws:iam::428396569305:instance-profile/EC2CotissFeedbackEditor ▼ [Create new IAM profile](#)

Hostname type [Info](#)

Don't include in launch template ▼

DNS Hostname [Info](#)

☐ Enable resource-based IPv4 (A record) DNS requests
☐ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Don't include in launch template ▼

Shutdown behavior [Info](#)

Don't include in launch template ▼

Not applicable for EC2 Auto Scaling

Stop - Hibernate behavior [Info](#)

Don't include in launch template ▼

Not applicable for Amazon EC2 Auto Scaling.

- Optional: Add CloudWatch monitoring if you would like to. It does not affect functionality but can be beneficial.
- Create your launch template.

Create your Auto Scaling Group and Load Balancer

If you plan on implementing Level Two without doing Level One – you do not need to complete this section.

1. Navigate to Auto Scaling Groups on the left-hand menu and select Create Auto Scaling Group.
2. Give your Auto Scaling Group a name and select the launch template that you have just created.


Name


Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.


Launch template [Info](#) [Switch to launch configuration](#)


Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.




[Create a launch template](#) 

Version





[Create a launch template version](#) 

3. Select Next
4. Under Network select the VPC you have created for this project and select all the Availability Zones and subnets which are displayed.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0f037f330b587393 (CotissFeedbackVPC)
168.16.0.0/16



[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets



ap-southeast-2a | subnet-0c86306d413bd2350
(CotissFeedbackSubnet01)
168.16.0.0/20



ap-southeast-2b | subnet-07b8e7e0fc0ba35ee
(CotissFeedbackSubnet02)
168.16.16.0/20



ap-southeast-2c | subnet-05d289cff0a48e067
(CotissFeedbackSubnet03)
168.16.32.0/20



[Create a subnet](#)

5. Select Next.
6. Under Load balancing, select Attach to a new load balancer.
7. A new section called Attach to a new load balancer should appear. Follow the following settings.
 - Keep Application Load Balancer selected.
 - Give your Load Balancer a name.
 - Under Load balancer scheme, select Internet-facing.
 - Under Listeners and routing, ensure that Default routing (forward to) has Create a target group as its option. Give the target group a name below that.

Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

☒ **Application Load Balancer**
HTTP, HTTPS

☐ **Network Load Balancer**
TCP, UDP, TLS

Load balancer name
Name cannot be changed after the load balancer is created.

CotissLoadBalancer

Load balancer scheme
Scheme cannot be changed after the load balancer is created.

☐ Internal

☒ Internet-facing

Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-0f037f3330b587393 [CotissFeedbackVPC](#)

Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

<input checked="" type="checkbox"/> ap-southeast-2b	subnet-07b8e7e0fc0ba35ee
<input checked="" type="checkbox"/> ap-southeast-2c	subnet-05d289cff0a48e067
<input checked="" type="checkbox"/> ap-southeast-2a	subnet-0c86306d413bd2350

Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol HTTP	Port 80	Default routing (forward to) Create a target group
New target group name An instance target group with default settings will be created.		
CotissTargetGroup		

8. Select Next.

9. Under group size, change the values to suit your needs. As our solution was never used on a large scale, the minimum, desired and maximum capacities were set to 1.
10. Under scaling policies, you can keep none selected or again choose to adjust the settings to your requirements. Defining your own policies for auto scaling would be beneficial.
11. Select Next.
12. Add notifications is optional and can be beneficial to check instance health but was not used in our case.
13. Select Next until you see the review screen.
14. Double check the settings to the settings that have been described alongside any changes that you may have made to fit your own needs.
15. Select Create Auto Scaling Group once you are happy that the settings are correct.

Creating an Endpoint for your DynamoDB

The instances that get launched under the load balancer and auto scaling group won't have a public IPv4 address. Hence, we will need to create an endpoint so it can access the DynamoDB Table.

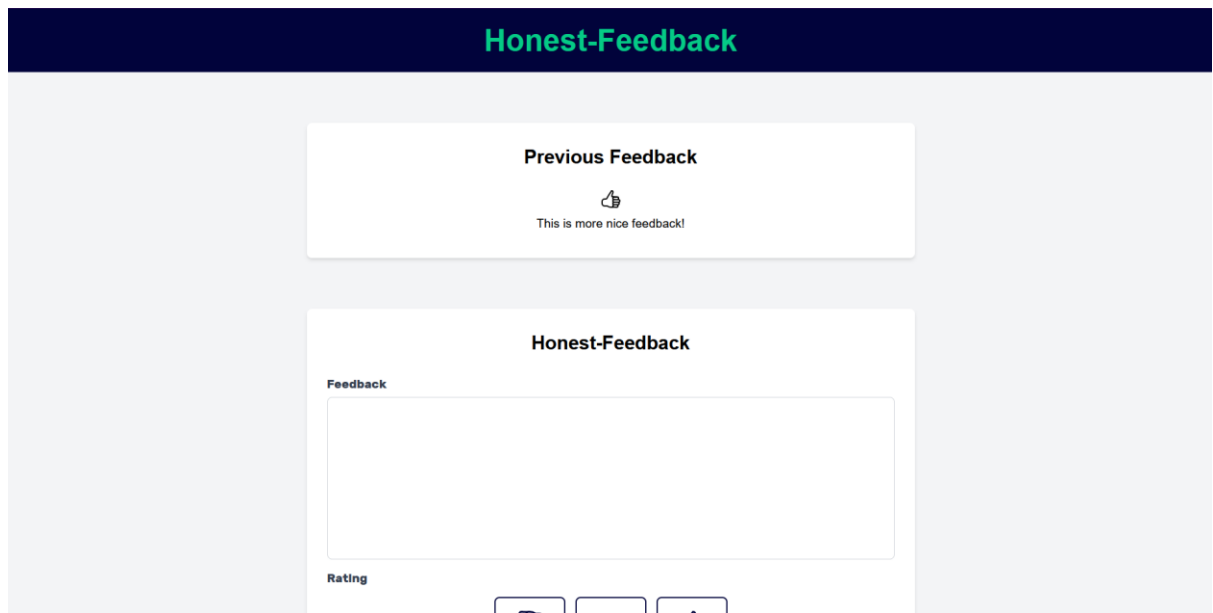
1. Make sure you have configured your environment for your AWS Account that you are doing this project on. If you have not configured this already, follow this [link](#) if you require more information.
2. Navigate to your VPC and copy the following details:
 - VPC ID
 - VPC's Route Table ID

vpc-0f037f3330b587393 / CotissFeedbackVPC			
<div> <div>Details</div> <div>Info</div> </div>			
VPC ID vpc-0f037f3330b587393	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-057a246bf6deafac0	Main route table rtb-04668041abcde8177	Main network ACL acl-037b8de9728e6883a
Default VPC No	IPv4 CIDR 168.16.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 428396569305	

3. Open your Command Line/cmd.
4. Run the following command in your command line replacing XXX... with your VPC ID and YYY with your Route Table ID.
 - **aws ec2 create-vpc-endpoint --vpc-id vpc-XXXXXXXX --service-name com.amazonaws.ap-southeast-2.dynamodb --route-table-ids rtb-XXXXXXX**

Check your Work

1. Navigate to Load Balancers in your EC2 Dashboard.
2. Copy its DNS and view the website.
3. If successful, your website should load like so.



The screenshot shows a web application titled "Honest-Feedback" with a dark blue header. The main content area is light gray and contains two white boxes. The top box, titled "Previous Feedback", displays a thumbs-up icon and the text "This is more nice feedback!". The bottom box, titled "Honest-Feedback", contains a text input field labeled "Feedback" and a rating section labeled "Rating" with three radio button options.

4. Enter your feedback and Rating and Submit.
5. Check your DynamoDB Table for your feedback and rating.

Step by Step Implementation Guide – Level Two

This level builds on Level One but with the addition of some new features:

- Serving all actions over HTTPS rather than HTTP.
- An addition of an S3 bucket which serves static content for the webpage (e.g. Images)
- The use of Elastic Beanstalk to create our EC2 instances with load balancers and auto scaling.
- Using a custom DNS for our webpage.

Creating a .zip for your Elastic Beanstalk

Elastic Beanstalk requires a .zip to launch and deploy your application. The files that need to be zipped up can be found [here](#). Note the new file from Level One – **composer.json**. This file specifies the SDK version which is used in the code that is provided.

Creating an Elastic Beanstalk Environment and Application

1. Navigate to your Elastic Beanstalk Dashboard.
2. Select Create a new Environment.
3. Keep Web Server Environment selected and click Select.
4. Give your application a name.
5. Scroll to platform and choose PHP as your platform.

Platform

☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ **Custom platform**
Platforms created and owned by you.

Platform
PHP ▼

Platform branch
PHP 8.1 running on 64bit Amazon Linux 2 ▼

Platform version
3.5.3 (Recommended) ▼

6. Select Upload your code and upload a local file. Choose the .zip file that you have created in the first step.
7. Select Configure more options.

8. Select Edit in Software.
9. Change Proxy Server to Apache and hit Save.
10. Select Edit in Capacity.
11. Set Environment Type to Load Balanced.
12. Adjust the Instances section's Min and Max values according to your needs.
13. Select Save.
14. Note the Virtual Machine Instance Profile in Security – we will need to edit this later.
15. Select Edit in Network.
16. Select the VPC that you have created for this project.
17. Enable all Load balancer subnets and Instance subnets and tick Public IP address.

Load balancer settings

Assign your load balancer to a subnet in each Availability Zone (AZ) in which your application runs. For a publicly accessible application, set **Visibility** to **Public** and choose public subnets.

Visibility
Make your load balancer internal if your application serves requests only from connected VPCs. Public load balancers serve requests from the Internet.

Public

Load balancer subnets

<input checked="" type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-southeast-2a	subnet-0c86306d413bd2350	168.16.0.0/20	CotissFeedbackSubnet01
<input checked="" type="checkbox"/>	ap-southeast-2b	subnet-07b8e7e0fc0ba35ee	168.16.16.0/20	CotissFeedbackSubnet02
<input checked="" type="checkbox"/>	ap-southeast-2c	subnet-05d289cff0a48e067	168.16.32.0/20	CotissFeedbackSubnet03

Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.

☒ **Public IP address**
Assign a public IP address to the Amazon EC2 instances in your environment.

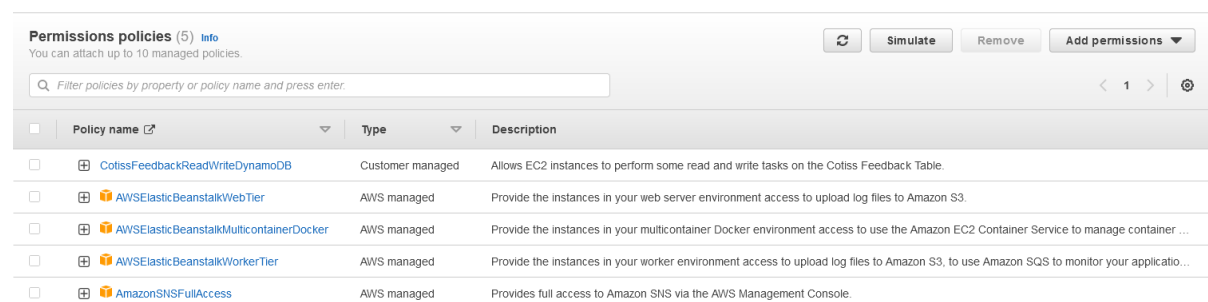
Instance subnets






<input checked="" type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-southeast-2a	subnet-0c86306d413bd2350	168.16.0.0/20	CotissFeedbackSubnet01
<input checked="" type="checkbox"/>	ap-southeast-2b	subnet-07b8e7e0fc0ba35ee	168.16.16.0/20	CotissFeedbackSubnet02
<input checked="" type="checkbox"/>	ap-southeast-2c	subnet-05d289cff0a48e067	168.16.32.0/20	CotissFeedbackSubnet03

18. Hit Save.
19. Create environment.

Adding our Policy to the EC2 Elastic Beanstalk Role

1. Navigate to your IAM dashboard.
2. Select Roles in the left-hand menu.
3. Find the Instance Profile that was mentioned in the previous section and click on it. It should be called **aws-elasticbeanstalk-ec2-role**.
4. Scroll to Permission Policies and select Add Permissions.
5. Select Attach Policies from the dropdown.
6. Check the policy that you have created for this project and select Attach policies.
7. Ensure that the policy has been added.



<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	 CotissFeedbackReadWriteDynamoDB	Customer managed	Allows EC2 instances to perform some read and write tasks on the Cotiss Feedback Table.
<input type="checkbox"/>	 AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server environment access to upload log files to Amazon S3.
<input type="checkbox"/>	 AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontainer Docker environment access to use the Amazon EC2 Container Service to manage container ...
<input type="checkbox"/>	 AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker environment access to upload log files to Amazon S3, to use Amazon SQS to monitor your applicatio...
<input type="checkbox"/>	 AmazonSNSFullAccess	AWS managed	Provides full access to Amazon SNS via the AWS Management Console.

Have an S3 Bucket serve your Static Data

Create an S3 Bucket.

1. Navigate to your S3 Dashboard.
2. Select Create bucket.
3. Give your bucket a name.
4. Select your AWS Region as your current region. (Sydney if you are following this documentation precisely.)
5. Keep ACLs disabled.
6. Unselect Block *all* public access and check the box acknowledging the risk.
7. Bucket Versioning is not important but feel free to enable if you desire.
8. Leave Default encryption as the default provided settings.
9. Create bucket.

Upload your Static Content – Logo Image

1. Select your newly created bucket.
2. Upload the [Cotiss Logo](#) used as the tab icon and on the webpage.

Registering a Domain for Route 53

Register or purchase a domain in Route 53 in the Route 53 Dashboard. This part is self explanatory and requires you to follow instructions provided on the webpage.

Creating a Hosted Zone

Creating a Hosted Zone

1. Navigate to your Route 53 Dashboard.
2. Select Hosted Zones on the left-hand menu.
3. Select Create Hosted Zone.
4. Enter your Domain Name from Route 53 earlier under Domain name.
5. Leave the rest of the settings as default and selected Create Hosted Zone.

Requesting a Certificate

1. Navigate to your Certificate Manager Dashboard.
2. Set your region to US East (N. Virginia). This is important for requesting certificates. Remember to change this back later.
3. Select Request a certificate. If you already have a certificate, you may import it but this documentation does not follow this route.
4. Keep Request a public certificate selected and select Next.
5. Under Fully qualified domain name (FQDN), add any FQDN that you would like to use.
 - E.g. *.yourdomain.com
6. Keep the remaining settings as the provided defaults and select Request.

Certificate status

Identifier

bdf2226a-31c6-4855-8c9c-879f2659ecbf

Status

Issued

ARN

arn:aws:acm:us-east-1:428396569305:certificate/bdf2226a-31c6-4855-8c9c-879f2659ecbf

Type

Amazon Issued

Domains (4)

Create records in Route 53

Export to CSV

<1>

Domain	Status	Renewal status	Type	CNAME name	CNAME value
gavinlim.link	<div><div></div><div>Success</div></div>	-	CNAME	<div><div></div><div>_d435017db0ebb8c8b41bc98668ecf59b.gavinlim.link.</div></div>	<div><div></div><div>.644e3bf6bee8ca7b073b91b7b55c11a1.fyfbssdptv.acm-validations.aws.</div></div>
cotiss.gavinlim.link	<div><div></div><div>Success</div></div>	-	CNAME	<div><div></div><div>_1285f89dd43c2966880005a1cbb8c7d.cotiss.gavinlim.link.</div></div>	<div><div></div><div>.21fabec5df4e4c4230979f0780b1946.fyfbssdptv.acm-validations.aws.</div></div>
*.cotiss.gavinlim.link	<div><div></div><div>Success</div></div>	-	CNAME	<div><div></div><div>_1285f89dd43c2966880005a1cbb8c7d.cotiss.gavinlim.link.</div></div>	<div><div></div><div>.21fabec5df4e4c4230979f0780b1946.fyfbssdptv.acm-validations.aws.</div></div>
*.gavinlim.link	<div><div></div><div>Success</div></div>	-	CNAME	<div><div></div><div>_d435017db0ebb8c8b41bc98668ecf59b.gavinlim.link.</div></div>	<div><div></div><div>.644e3bf6bee8ca7b073b91b7b55c11a1.fyfbssdptv.acm-validations.aws.</div></div>

7. Click the Create records in Route 53 Button.
8. Select the checkbox next to all the options available and then select Create records.
9. Your Certificate should get approved shortly.

Creating CloudFront Distributions

Create a CloudFront Distribution for your Elastic Beanstalk Environment

1. Navigate to CloudFront and select Create Distribution
2. Under Origin Domain, select your Elastic Beanstalk Web Address. Feel free to go back to your Elastic Beanstalk Dashboard and find this if you need to.
3. Change Origin Access to Origin Access Control Settings.
4. Create a new control setting – give it a name and select create, nothing needs to be changed.
5. Give a name to your Distribution.
6. Scroll to Default Cache Behaviour.
7. Change Viewer Protocol Policy under Viewer to Redirect HTTP to HTTPS
8. Change Allowed to HTTP methods to include POST (i.e. GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE). Leave OPTIONS unchecked.
9. Under Cache Key and Origin Requests, keep the recommended setting of Cache policy and origin request policy.
10. Change Cache Policy from CachingOptimized to CachingDisabled. This is required for the randomly generated feedback to be randomized each time and not the same cached one.

Viewer

Viewer protocol policy

☐ HTTP and HTTPS

☒ Redirect HTTP to HTTPS

☐ HTTPS only

Allowed HTTP methods

☐ GET, HEAD

☐ GET, HEAD, OPTIONS

☒ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

☐ OPTIONS

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

☒ No

☐ Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

☒ Cache policy and origin request policy (recommended)

☐ Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized

Default policy when CF compression is enabled

Recommended for S3 origins

⌵

[Create policy](#) [View policy](#)

Origin request policy - optional

Choose an existing origin request policy or create a new one.

Select origin policy

⌵

[⌵](#)

11. Scroll to settings and select Add item under Alternate domain name (CNAME).
12. Scroll until Alternate Domain Name (CNAME) and add item.
13. Add in the domain name you wish to use for your webpage. (e.g. cotiss.gavinlim.link)
14. Under Custom SSL Certificate, select the Certificate that you requested previously.

Settings

Price class [Info](#)

Choose the price class associated with the maximum price that you want to pay.

☒ Use all edge locations (best performance)

☐ Use only North America and Europe

☐ Use North America, Europe, Asia, Middle East, and Africa

AWS WAF web ACL - optional

Choose the web ACL in AWS WAF to associate with this distribution.

Choose web ACL

Alternate domain name (CNAME) - optional

Add the custom domain names that you use in URLs for the files served by this distribution.

cotiss.gavinlim.link

Remove

Add item

To add a list of alternative domain names, use the [bulk editor](#).

Custom SSL certificate - optional

Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

gavinlim.link (bdf2226a-31c6-4855-8c9c-879f2659ecbf)

gavinlim.link [Request certificate](#)

15. Create Distribution.

Create Distribution for your Static Content Bucket

1. Select Create Distribution.
2. Choose your S3 bucket that you previously created as your Origin Domain.
3. Select Origin Access Control Settings under Origin Access.
4. Choose the Control Setting you created for the Elastic Beanstalk Distribution for the Origin Access Control setting.
5. Change Viewer Protocol Policy under Viewer to Redirect HTTP to HTTPS.

Path pattern [Info](#)

Default (*)

Compress objects automatically [Info](#)

☐ No

☒ Yes

Viewer

Viewer protocol policy

☐ HTTP and HTTPS

☒ Redirect HTTP to HTTPS

☐ HTTPS only

Allowed HTTP methods

☒ GET, HEAD

☐ GET, HEAD, OPTIONS

☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

☒ No

☐ Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

☒ Cache policy and origin request policy (recommended)

☐ Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized Recommended for S3 origins ▼

Default policy when CF compression is enabled

[Create policy](#) [View policy](#)

6. Select Add Item under Alternate Domain Name (CNAME).
7. Add in the domain name you wish to use for your static bucket (eg. static.cotiss.gavinlim.link)
8. Under Custom SSL Certificate, choose the Certificate that you requested previously.
9. Create Distribution.
10. A blue banner may appear, click Copy Policy if so and then Go to S3 bucket permissions to update policy.
11. Click Edit on bucket policy.
12. Paste the policy that you have copied and save changes.

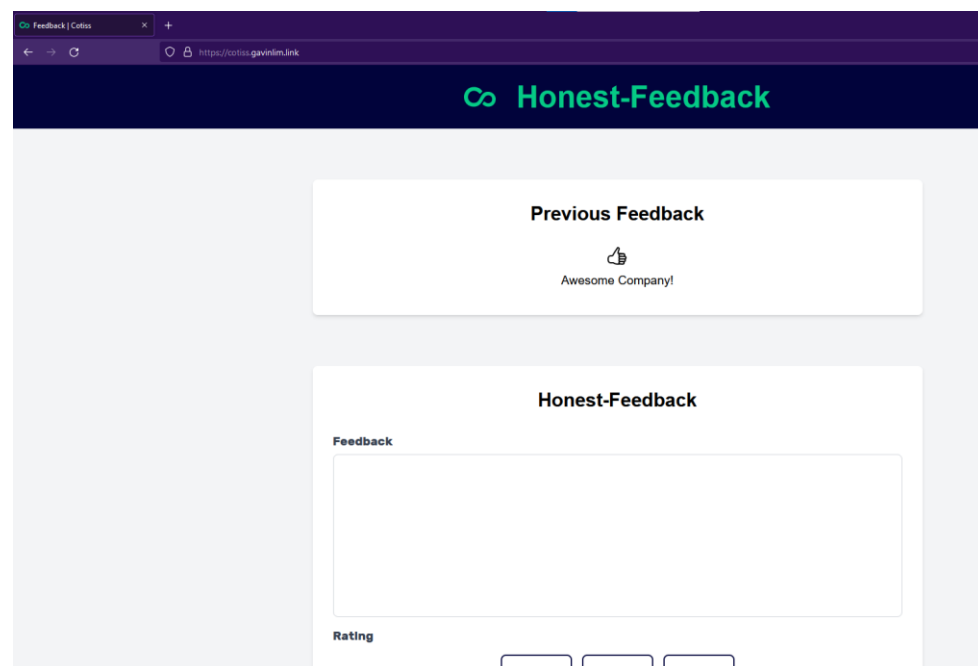
Adding your CloudFront Distributions as Records

1. Navigate to Route 53.
2. Selected Hosted Zones in the left-hand menu.
3. Select your Hosted Zone.
4. Select Create Record.
5. Add in the alternate domain name for your Elastic Beanstalk Instance for Record Name.
6. Keep record type as A.
7. Add the CloudFront Distribution Domain Name for your Elastic Beanstalk Cloudfront Distribution into the Value box. (e.g. d1ear5aids8b2n.cloudfront.net)
8. Create Record.
9. Repeats steps 5 to 8 for your Static Bucket.

<input type="checkbox"/>	Record name ▾	Type ▾	Routin... ▾	Differ... ▾	Value/Route traffic to ▾
<input type="checkbox"/>	gavinlim.link	NS	Simple	-	ns-521.awsdns-01.net. ns-137.awsdns-17.com. ns-1721.awsdns-23.co.uk. ns-1110.awsdns-10.org.
<input type="checkbox"/>	gavinlim.link	SOA	Simple	-	ns-521.awsdns-01.net. awsdns-hostmaster.amazon.com. 1 7200 900 ...
<input type="checkbox"/>	_d435017db0ebb8c8b41bc98...	CNAME	Simple	-	_644e3bf6bee8ca7b073b91b7b55c11a1.fyfbssdptv.acm-validations...
<input type="checkbox"/>	cotiss.gavinlim.link	A	Simple	-	d3lfxcg34jm6vz.cloudfront.net.
<input type="checkbox"/>	_31285f89dd43c2966880005a...	CNAME	Simple	-	_2bfabec5dfd4e4c4230979f0780b1946.fyfbssdptv.acm-validations.a...
<input type="checkbox"/>	static.cotiss.gavinlim.link	A	Simple	-	dtudra3gk64o4.cloudfront.net.

Check your work

1. Visit your domain name that you used for your webpage and CloudFront Distribution.
2. Check to see if your webpage serves data over HTTPS and has the additional images.



Cost Analysis

The following costs are listed in USD. Prices are current as of 15th January 2023. This calculation is done assuming 8760 hours in a year. You can view an in-depth analysis by visiting the links for each respective level.

Level One

[Level One Full Cost Analysis Link.](#)

Service	Cost/Month	Cost/Year	Upfront Cost	Number of Instances	Total
EC2 (t2.micro)	\$14.31	\$171.72	\$0.00	2	
DynamoDB	\$14.00	\$168.00	\$205.20	1	
Elastic Load Balancer	\$21.32	\$255.84	\$0.00	1	
Total	\$49.63	\$595.56	\$205.20		\$800.74

Level Two

[Level Two Full Cost Analysis Link.](#)

Service	Cost/Month	Cost/Year	Upfront Cost	Number of Instances	Total
EC2 (t2.micro)	\$14.31	\$171.72	\$0.00	2	
DynamoDB	\$14.00	\$168.00	\$205.20	1	
Elastic Load Balancer	\$21.32	\$255.84	\$0.00	1	
S3 Bucket	< \$0.01	< \$0.01	\$0.00	2	
Cloudfront	\$0.40	\$4.80	\$0.00	1	
Route 53	\$0.54	\$6.48	\$0.00	1	
Total	\$50.57	\$606.89	\$205.20		\$812.02