

SCHOOL OF COMPUTING (SOC)



# IOT CA2 Step-by-step Tutorial

DIPLOMA IN BUSINESS INFORMATION TECHNOLOGY  
DIPLOMA IN INFORMATION TECHNOLOGY  
DIPLOMA IN INFOCOMM SECURITY MANAGEMENT

**ST0324 Internet of Things (IOT)**

**Date of Submission:**

**Prepared for:**

**Class:**

**Submitted by:**

**Student ID    Name**

# Table of Contents

Section 1 Overview of project .....	2
A. Where we have uploaded our tutorial .....	2
B. What is the application about? .....	2
C. How does the final RPI set-up looks like? .....	2
D. How does the web or mobile application look like? .....	4
E. System architecture of our system .....	6
F. Quick-start guide (Readme first).....	6
Section 2 Hardware requirements .....	7
Hardware checklist.....	7
Hardware setup instructions .....	7
Fritzing Diagram .....	7
Section 3 Software Requirements.....	8
Software checklist .....	8
Software setup instructions.....	8
A. Setup Web Server and Database in VPC.....	9
B. Setup Lambda functions .....	14
C. Setup MQTT Broker .....	18
D. Setup AWS Rekognition with Custom Labels.....	21
Section 4 Task List.....	22
Section 5 References .....	22

# Section 1

## Overview of project

### A. Where we have uploaded our tutorial

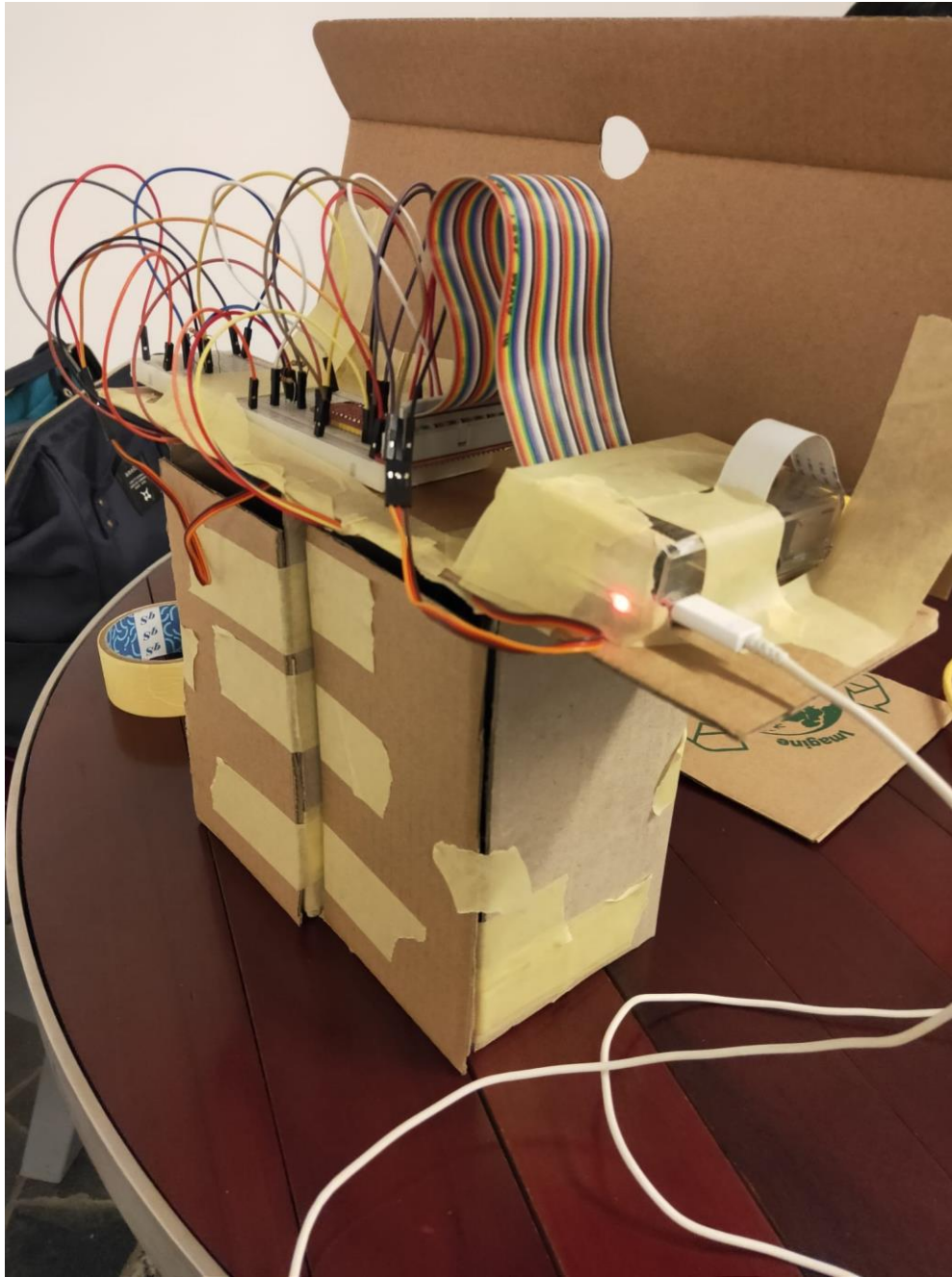
[REDACTED]

### B. What is the application about?

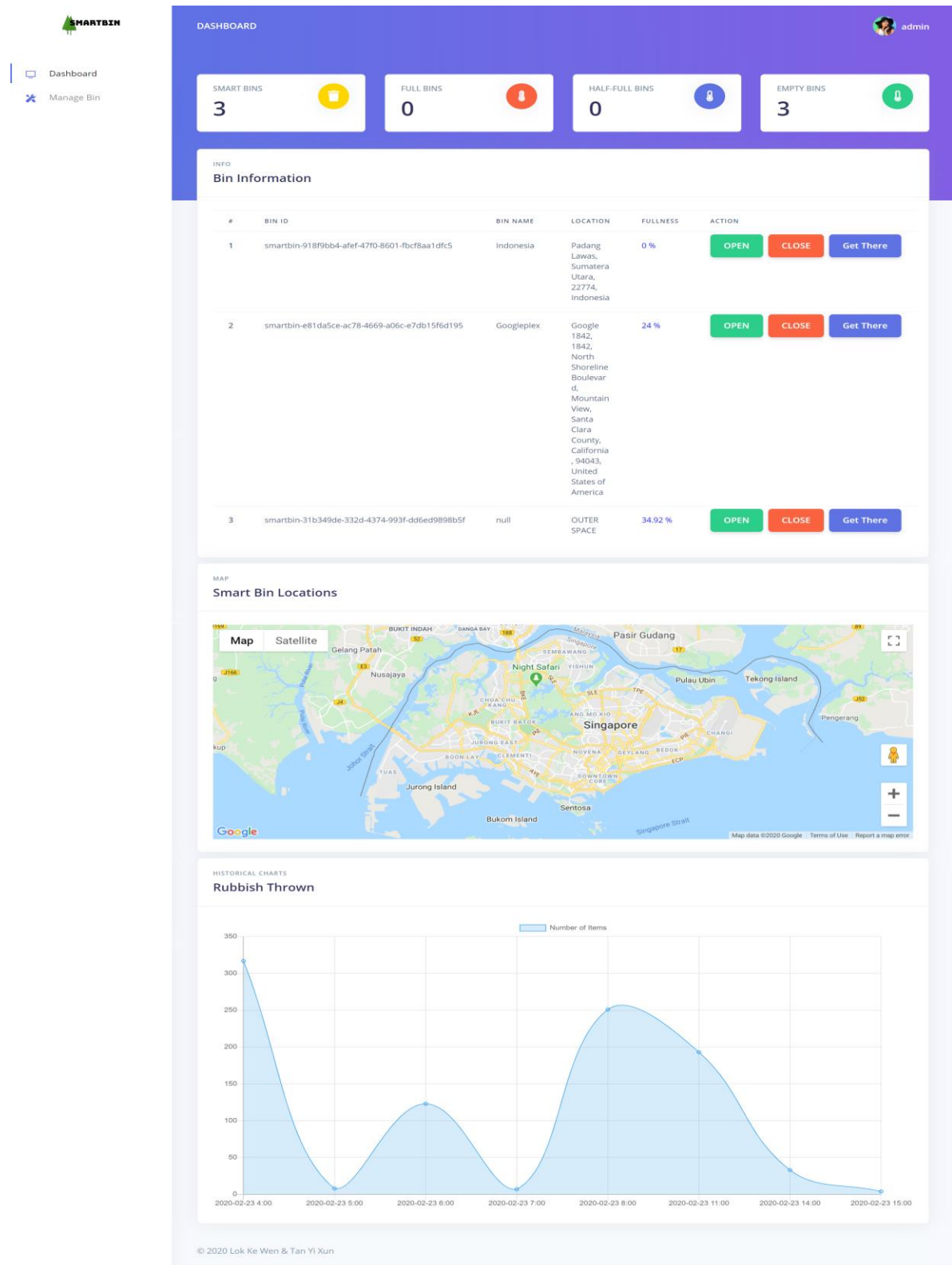
The application is called SmartBin. The purpose of this application is to allow easier management of bins around the country. For this case, the targeted end user of this application will be the government. This application is meant as a one stop interface which provides real-time and historical data of bins around Singapore using sensors, which will provide information such as bin fullness and location to enable garbage collectors to know when certain bins around Singapore are full so they can clear. This would allow the government and the garbage collectors to be more efficient in their job and reduce overhead of driving around and constantly checking empty bins.


### C. How does the final RPI set-up looks like?





## D. How does the web or mobile application look like?





[Dashboard](#)[Manage Bin](#)

MANAGE BIN

admin

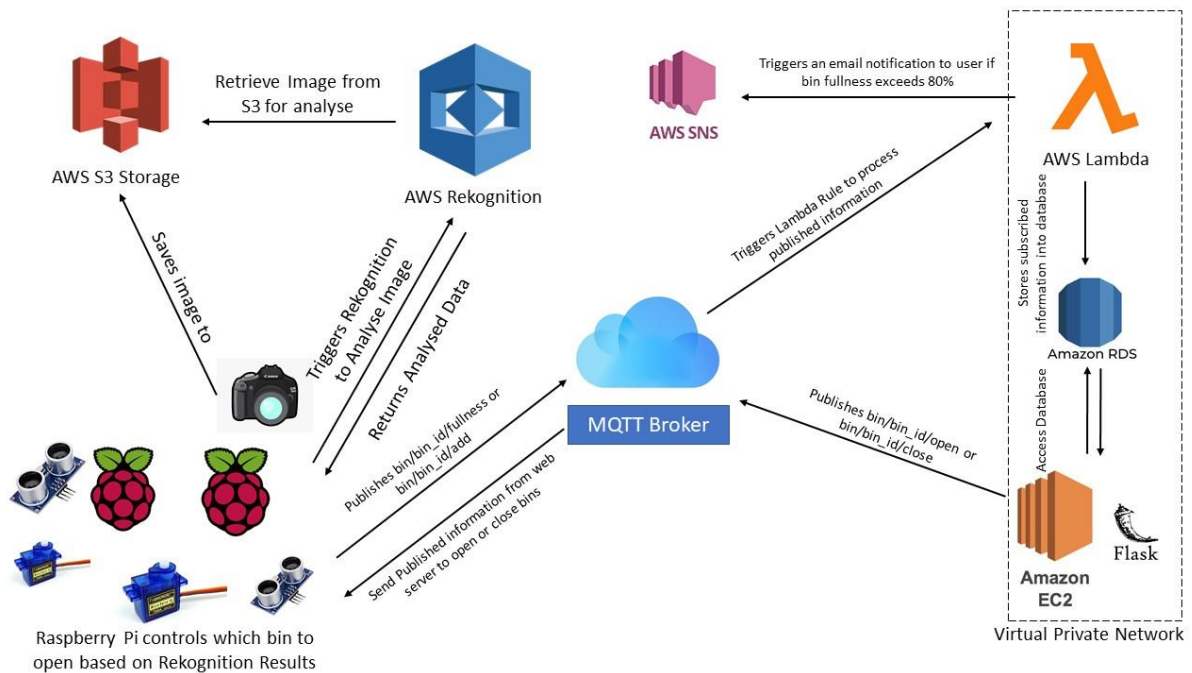
MANAGEMENT PANEL

Bin Information

#	BIN ID	BIN NAME	LOCATION	FULLNESS	ACTION
1	smartbin-31b349de-332d-4374-993f-dd6ed9898b5f	None	OUTER SPACE	34.92 %	<a href="#">Update</a> <a href="#">Delete</a>
2	smartbin-e81da5ce-ac78-4669-a06c-e7db15f6d195	Googleplex	Google 1842, 1842, North Shoreline Boulevard, Mountain View, Santa Clara County, California, 94043, United States of America	24.0 %	<a href="#">Update</a> <a href="#">Delete</a>
3	smartbin-918f9bb4-afef-47f0-8601-fbcf8aa1dfc5	Indonesia	Padang Lawas, Sumatera Utara, 22774, Indonesia	0.0 %	<a href="#">Update</a> <a href="#">Delete</a>

© 2020 Lok Vir World & Tan T' Hoon

## E. System architecture of our system



## F. Quick-start guide (Readme first)

- 1) First, setup the hardware as shown in **Section 2** fritzing diagram.
- 2) Set up all AWS services as shown in **Section 3**.
- 3) Copy the Web Server folder in the source code folder to EC2 web server.
- 4) Make sure you have installed all required software as stated in **Section 3 Software Checklist**.
- 5) Run the **server.py** file.
- 6) Copy the Raspberry Pi folder to Raspberry Pi.
- 7) Make sure you have installed all required software as stated in **Section 3 Software Checklist**.
- 8) Run the **smartbin.py** file.



## Section 2

### Hardware requirements

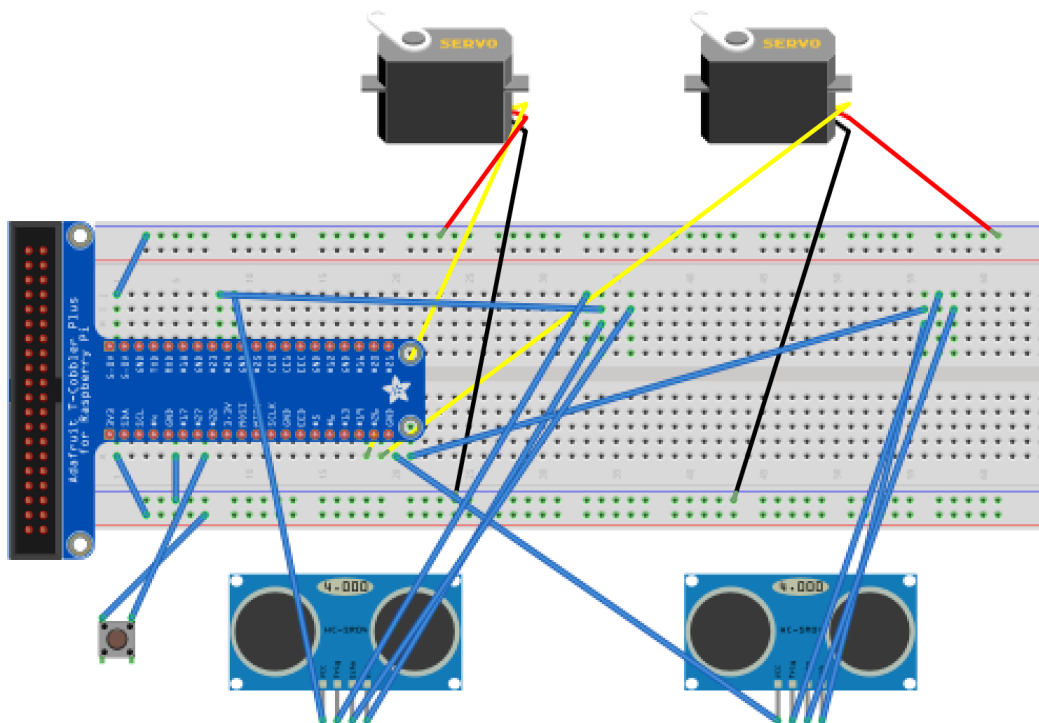
#### Hardware checklist

- a) 2 Raspberry Pi
- b) 2 Cameras
- c) 4 Servo Motors
- d) 4 Ultrasonic Sensors
- e) 2 Buttons

#### Hardware setup instructions

Make sure that the PiCamera is connected to its designated slot on Raspberry Pi and setup the other hardware as shown below.

#### Fritzing Diagram





## Section 3

# Software Requirements

### Software checklist

1. PIL
2. Picamera
3. Numpy
4. Botocore
5. AWSIoTPythonSDK
6. Uuid
7. RPi
8. Mysql.connector
9. import json
10. import boto3
11. geopy

### Software setup instructions

All the library required for this application except for OpenCV can be installed using the command **"sudo pip install <library package name>"**, whereby the library package name is provided in the **Software Checklist**. This section will focus on setting up essential cloud services on Amazon Cloud Services, and also ways to train your custom model using AWS Rekognition Custom Labels.

When using some python programmes that utilises AWSCLI library to connect to your AWS account services, you will need to copy your account session from Vocareum once in a while. This is due to the fact that AWS Educate account is restricted from creating a key pair for users created using IAM. To overcome this issue, login to your AWS Educate account, select "Go to classroom". When the Vocareum website has been loaded, click "Account Details". You will then need to click the "Show" button beside AWS CLI, and copy the shown credentials to "~/.aws/credentials". The credentials that you have just copied will serve as temporary session for AWS CLI to connect to your AWS Services. Once the session has expired, you will need to re-do the steps all over again.

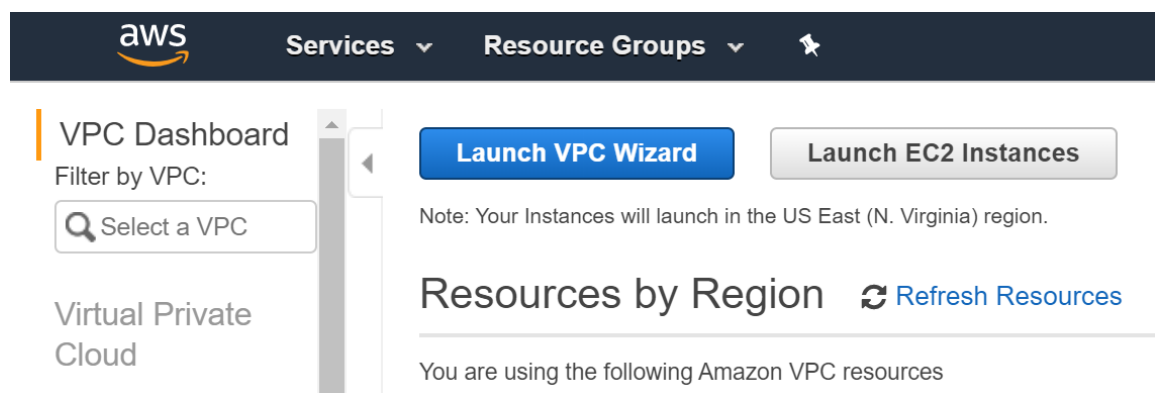
Google Maps API key is required by the frontend web server to show directions to the bin and display locations of all smart bins installed. To get the API key, you will need to sign up for a Google Cloud Account using your google account and enter your payment information for verification. You can generate Google Maps API key after assigning a billing account to your project. The API needed are Google Maps Embed API and Google Maps JavaScript API.

## A. Setup Web Server and Database in VPC

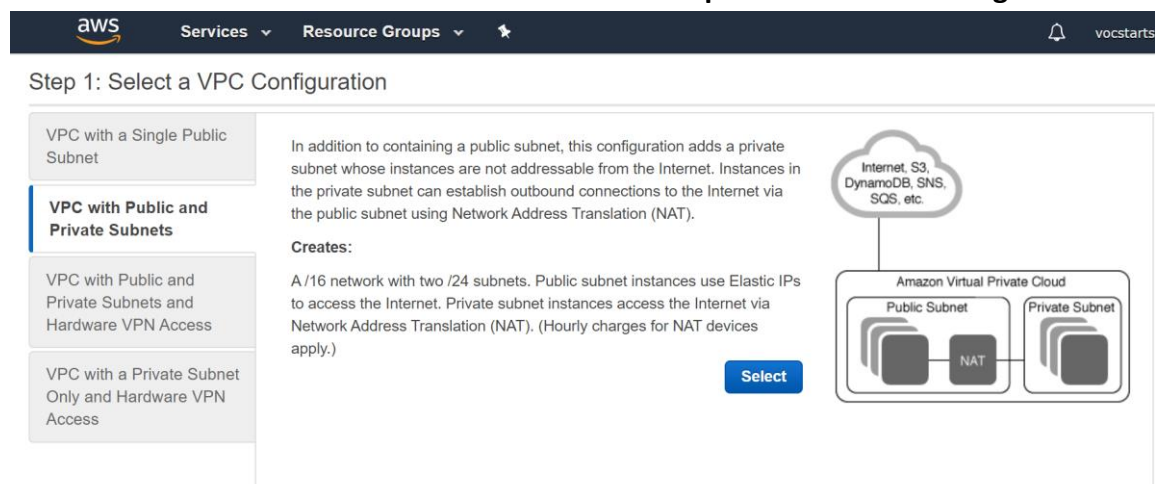
### Task

- a) Before creating web server (EC2) and database (RDS), a Virtual Private Cloud (VPC) with both public and private subnets, and corresponding security groups must be created. VPC is responsible for segmenting the network to only allow servers in the same VPC with the privileges to access the database. The official link for setting up the VPC can be found in this link.  
([https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateVPC.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html)) Alternatively, you can follow my step-by-step guide below.

- b) Go to Amazon VPC console. (<https://console.aws.amazon.com/vpc/>) On the left navigation pane, choose VPC Dashboard. And then, click "Launch VPC Wizard".



Choose "VPC with Public and Private Subnets" for **Step 1: Select a VPC Configuration**.



For **Step 2: VPC with Public and Private Subnets** page, enter the values as follow:

## Task

**IPv4 CIDR Block : 10.0.0.0/16**

**IPv6 CIDR Block : No IPv6 CIDR Block**

**VPC Name : MySmartBin-vpc**

**Public subnet's IPv4 CIDR : 10.0.0.0/24**

**Availability Zone : us-east-1a**

**Public Subnet Name : Public Subnet**

**Private Subnet's IPv4 CIDR : 10.0.1.0/24**

**Availability Zone : us-east-1a**

**Private Subnet Name : Private Subnet**

**Instance type : t2.small (If you did not see any instance, choose "Use NAT instance instead")**

**Key Pair Name : No key pair**

**Service endpoint : Skip this field**

**Enable DNS Hostnames : Yes**

**Hardware Tenancy : Default**

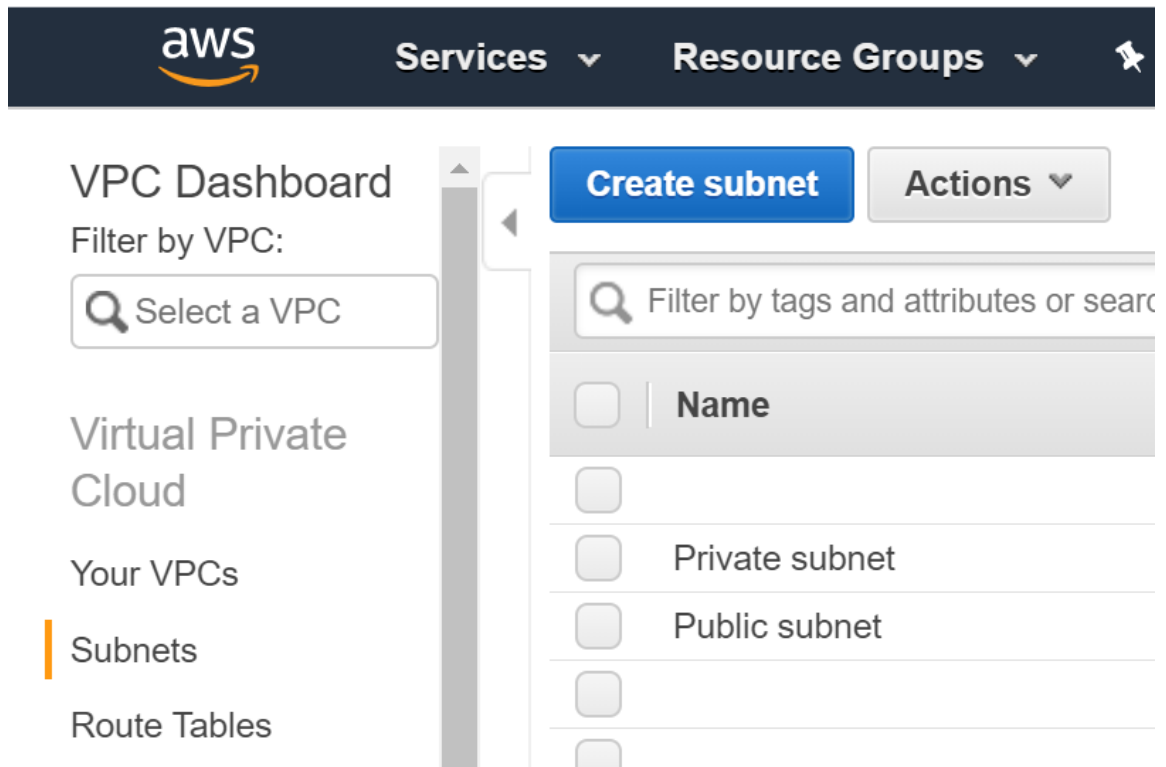
After all fields are filled as above, click **"Create VPC"**

Your VPC will be created in a few minutes.

- c) A DB subnet group is required for a DB instance to be used in a VPC. In order to create a DB subnet group, there must be at least two public subnets or two private subnets available. As DB instance is intended to be used in private subnet, you will now create an extra private subnet.

Go to Amazon VPC console. (<https://console.aws.amazon.com/vpc/>) On the left navigation pane, choose VPC Dashboard, choose Subnets, and then, click "Create subnet".

## Task



On the **Create Subnet** page, enter the values as follow:

**Name Tag : Private Subnet for DB Instance**

**VPC : Choose the VPC that you have created in step b.**

**Availability Zone : us-east-1b**

**IPv4 CIDR Block : 10.0.2.0/24**

Once you have done with all the steps above, click **“Create”** to create the subnet.

- d) Next, you will need to create and configure a VPC security group for your web server (will be created later). This security group will allow your web server to receive web request from external machines and connect to database at the same time.

Go to Amazon VPC console. (<https://console.aws.amazon.com/vpc/>) On the left navigation pane, choose VPC Dashboard, choose Security Groups, and then, click “Create Security Group”.

## Task

In the “**Create security group**” page, enter values as follow:

**Security group name : MySmartBinWebServer-securitygroup**

**Description : VPC Security Group for Web Server**

**VPC : The VPC you have just created in Step b.**

After that, click “**Create**” to create the security group.

You can access and edit your inbound rule by clicking the created security group, click **Inbound Rules**, and click **Edit Rule**. To allow remote access and web request to your web server, set your security inbound rule as follow:

**Rule 1:**

**Type : SSH**

**Source : 0.0.0.0/0**

**Rule 2:**

**Type : HTTP**

**Source : 0.0.0.0/0**

Click “**Save rules**” and close when you are done.

Now, you will need to create a VPC security group for DB instance. The steps are the same as above except the values for **Create Security Group** and **Inbound Rule** is slightly different. The values are as follow:

**Security group name : MySmartBinDBInstane-securitygroup**

**Description : VPC Security Group for DB Instance**

**VPC : The VPC you have just created in Step b.**

**Rule 1:**

**Type : MySQL/Aurora**

**Source : The identifier of MySmartBinWebServer-securitygroup (sg-xxxxxxx)s**

- e) Since the additional private subnet has been created in the previous step, you can now create a DB subnet group specifically for your DB instance.

## Task

DB subnet group can be created using Amazon RDS console (<https://console.aws.amazon.com/rds/>) since we are using relational database, MySQL.

In the navigation pane of the Amazon RDS console, choose **Subnet Groups**. Then, **Create DB Subnet Group**.

On the **Create Security Group** page, enter the values as follow:

**Name :** MySmartBin-db-subnet-group

**Description :** MySmartBin DB Subnet Group

**VPC :** Choose the VPC you have created in Step b.

In the **Add subnets** section, choose **Add all the subnets related to this VPC**. Finally, click **"Create"**.

- f) Since you have setup all the required security groups and subnets, it is time to setup RDS DB Instance. In this case, you will deploy an MySQL DB instance.

First, open AWS RDS console. (<https://console.aws.amazon.com/rds/>) On the left navigation pane, choose **Databases**. And then, **Create database**.

For the **Create Database** page, please refer to **Step 5 to Step 14** for new console in this link:

[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateDBInstance.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html)

Do take note that you will need to enter values of some field based on the one specified below:

**DB instance identifier :** mysmartbin-db-instance

**Master username :** mysmartbin\_user

**Master password :** 1qwer\$#!

**Confirm password :** 1qwer\$#!

**Virtual Private Cloud (VPC) :** The one you created in step b

## Task

**Subnet Group : mysmartbin-db-subnet-group**

**VPC Security Groups : MySmartBinDBInstane-securitygroup**

- g) When the DB Instance is up and running, it is time for you to set up a web server. This web server will be hosting a website that allows authorised users to control and monitor all the smart bins connected to the network.

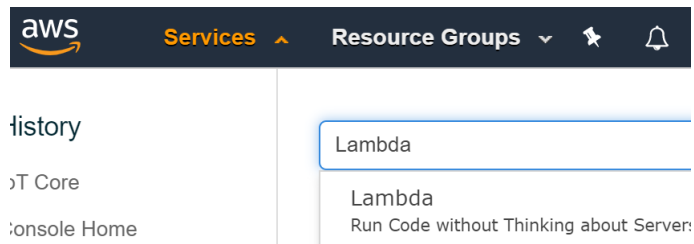
The web server will be running using EC2 services. To set up the EC2 instances, refer to **Step 1 to Step 16** of the link below:

[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateWebServer.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html)

## B. Setup Lambda functions

## Task

- a) Click on services and search for Lambda



Create a new lambda function and enter "SmartBinAddBin"

Function name

Enter a name that describes the purpose of your function.

SmartBinAddBin

 A function with that name already exists.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

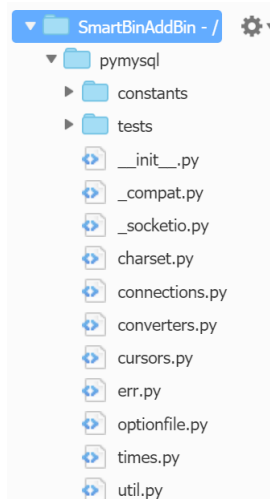
Choose the language to use to write your function.

Python 2.7



## Task

After creating the function, add the following code to the lambda function, and fill in the rds host with the one you created. Use a zip file with pymysql module files to be uploaded to the lambda function directory.



Also place the lambda function in the VPC as such.

### VPC

#### Virtual Private Cloud (VPC) [Info](#)

Choose a VPC for your function to access.

vpc-0cd2a93862f0cd4a0 (10.0.0.0/16) | MySmartBinWebServer-vpc

#### Subnets

Select the VPC subnets for Lambda to use to set up your VPC configuration. Format: "subnet-id (cidr-block) | az name-tag".

subnet-0ddec2c9de0ad4f35 (10.0.2.0/24) | us-east-1b Private Subnet for DB Server

subnet-02d2030a1732ee7a6 (10.0.1.0/24) | us-east-1a Private subnet

#### Security groups

Choose the VPC security groups for Lambda to use to set up your VPC configuration. Format: "sg-id (sg-name) | name-tag". The table below shows the inbound and outbound rules for the security groups that you chose.

sg-0a3a415b4e83443f6 (MySmartBinWebServer-securitygroup)

**i** When you connect a function to a VPC in your account, it does not have access to the internet unless your VPC provides access. To give your function access to the internet, route outbound traffic to a NAT gateway in a public subnet. [Learn more](#)

Click save.

## Task

Repeat the steps above for the second lambda function “SmartBinStoreFullness”, only difference will be the code for the lambda\_function shown below.

Click save.

b) Under services, search for IAM, and then click on Roles.

### Identity and Access Management (IAM)

- Dashboard
- ▼ Access management
  - Groups
  - Users
  - Roles**

For the lambda execution roles automatically created when the lambda function is created, it will be shown here.

[SmartBinAddBin-role-ehspc7a9](#)

[SmartBinStoreFullness-role-k7gamkww](#)

In the SmartBinAddBin role, add permissions for it to be able to access ec2 and rds. In the SmartBinStoreFullness role, add permissions for it to be able to access ec2 and sns. This would allow the lambda functions to be able to make the necessary API calls.

Additionally, for the SmartBinStoreFullness lambda function to be able to make SNS calls from within the VPC, it is required to add an endpoint. Therefore, in the services, search for VPC, then click on endpoint.

## Task

## VPC Dashboard

Filter by VPC:

 Select a VPC

## Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Click on create endpoint. Create the endpoint as follows

## Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.

An interface endpoint is powered by [PrivateLink](#), and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.

A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

- Service category**
- ☐ AWS services
  - ☒ Find service by name
  - ☐ Your AWS Marketplace services

**Service Name** Enter private service name and verify. ⓘ

 com.amazonaws.us-east-1.sns

Service name found.

 Verify

**VPC\*** vpc-0cd2a93862f0cd4a0

**Subnets**

Filter by attributes			
vpc-0cd2a93862f0cd4a0	10.0.0.0/16	available	MySmartBinWebServer-vpc
vpc-c56f30bf	172.31.0.0/16	available	

Select the following security group. Then click create endpoint.

## Task

Security group sg-0a3a415b4e83443f6 [Create a new security group](#) 

Select security groups ▲

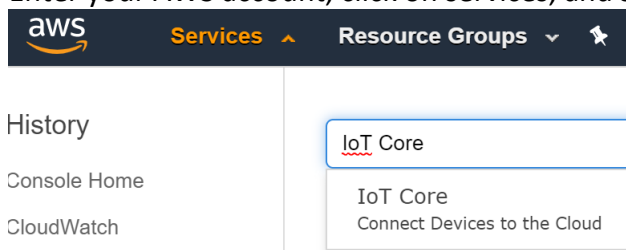
Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Group ID	Group Name
<input checked="" type="checkbox"/>	sg-0a3a415b...	MySmartBinWebServer-securitygroup

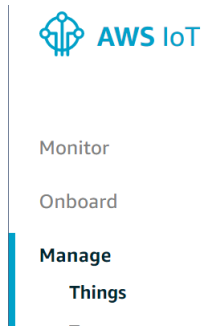
## C. Setup MQTT Broker

## Task

- a) Enter your AWS account, click on services, and search for IoT Core.



On the left panel, click on Manage, and then Things. After that, click Create on the right side of the page.



Click on Create a single thing. Name it “SmartBin”, then click next. After that, click on Create certificate.

## Task

### Register a single AWS IoT thing

Create a thing in your registry

Create a single thing

Download the following 3 files shown below, and a root CA file from the link below the 3 files. Afterwards, press Activate, and Done.

In order to connect a device, you need to download the following:

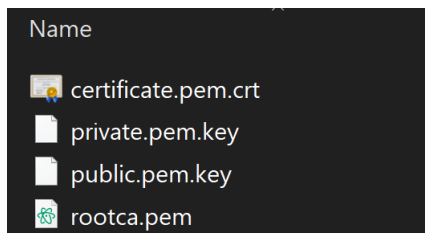
A certificate for this thing	661a74c1fe.cert.pem	<a href="#">Download</a>
A public key	661a74c1fe.public.key	<a href="#">Download</a>
A private key	661a74c1fe.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

Activate

Save the files to the certs folder and rename as such



- b) Under things click on the SmartBin, then take note of the API endpoint which would be used to put into the smartbin.py file mentioned in Section 4.

Back to the AWS IoT dashboard, click on Act, then Rules in the left side panel to create the rules. Click on create. Create the first rule "SmartBinAddBinRule" as such.

## Task

## Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a rule is triggered (e.g., write data to a DynamoDB table or invoke a Lambda function).

Name

SmartBinAddBinRule

Description

SmartBinAddBinRule



## Rule query statement

SELECT <Attribute> FROM <Topic Filter>  
WHERE temperature > 50. To learn more

```
1 SELECT * FROM 'bin/+/add'
```

Add action and follow as such. Then click create rule.



Send a message to a Lambda function

LAMBDA

We'll set [the permissions](#) on the Lambda function for you.

\*Function name

SmartBinAddBin

Follow similar steps for the second rule “SmartBinFullnessRule” except certain stuff as shown below.

Rule query statement

The source of the messages you want to process

```
SELECT * FROM 'bin/+/fullness'
```

## Task



Send a message to a Lambda function

LAMBDA

We'll set the [permissions](#) on the Lambda function for you.

\*Function name

SmartBinStoreFullness

## D. Setup AWS Rekognition with Custom Labels

### Task

- a) AWS Rekognition offers advanced image recognition capabilities to its user even the user is not tech-savvy. However, you are going to retrain your own model using AWS Rekognition Custom Labels as this project uses image recognition specifically for trash or recyclable item identification. It helps the smart bin to identify which bin it should open for each type of trash.

The custom labels can be trained and use by referring to the following guide:

<https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/gs-step-train-model.html>

The datasets used by the team to train the model can be obtained from:

<https://raw.githubusercontent.com/garythung/trashnet/master/data/dataset-resized.zip>



## Section 4

### Task List

A table listing members names and the parts of the assignment they worked on

Name of member	Part of project worked on	Contribution percentage
----------------	---------------------------	-------------------------

## Section 5

### References

<https://docs.aws.amazon.com/sns/latest/dg/sns-vpc-tutorial.html>  
<https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/>  
<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>  
<https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/gs-send-data.html>  
[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateDBInstance.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html)  
<https://docs.aws.amazon.com/AmazonS3/latest/gsg/CreatingABucket.html>  
[https://github.com/awsdocs/amazon-rekognition-developer-guide/blob/master/doc\\_source/images-s3.md](https://github.com/awsdocs/amazon-rekognition-developer-guide/blob/master/doc_source/images-s3.md)  
<https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/gs-step-train-model.html>  
<https://towardsdatascience.com/how-to-build-an-image-classifier-for-waste-sorting-6d11d3c9c478>  
<https://developers.google.com/maps/documentation/javascript/markers>  
<https://console.aws.amazon.com/rekognition/custom-labels#/projects/MySmartBin-Custom-Label-Training>  
<https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/Rekognition%20Custom%20Labels.pdf>

-- End of CA2 Step-by-step tutorial --