An Abstract or Executive Summary (one page)

Table of Contents

1. Problem Description:

1.5 Introduction of this project:

The domain of using the Internet is continuously shifting and evolving. In the past, Internet communication was human to human based. With the advancement of technologies, this paradigm has been shifted into the machine to machine communication and introduced the emerging concept of the Internet of Things (IoT) connecting anyone or anything at any time and any place through the Internet (Khan et al. 2012). The central concept is adding sensors with anything that allow the communication among them through the Internet so that they can always be connected. Those sensors mainly generate data where the internet carry those data as the carrier and virtualize resource concept of cloud provide all computational and processing power to make the decision and provide astonishing services by IoT applications. With the advent of cloud facilities, so many IoT applications evolved within a short time around the world such as environmental protection, health monitoring, intelligent search, smart traffic, smart city and so on. Moreover, the application of networked home appliances is increasing over time and with the assistance of those appliances and cloud services we aim to design our project which is a Smart Kitchen.

We plan to append intelligence into conventional machines such as refrigerator, coffee machine, gas cylinder, water tap and so on by adding sensors and collaborating them with cloud services. The smart kitchen project aims to provide online-based services to reduce some kitchen-centric hassle. We intend to provide services including notification for the groceries (egg, bread, milk, meat, fish, vegetables) if they finish instantly, generating weekly grocery shopping list, providing price matching web services (suggesting the lowest price and the name of the shop), tracking daily water and gas consumption and detecting gas leakage or high temperature in the kitchen.

This project aims to contribute to any other massive IoT application domain such as a smart city. In that case, a smart kitchen can be considered a small initiative with limited scope. However, this project emphasises more on the integration of cloud services to promote new featured IoT application to increase the level of human comfort, safety and track resource consumption to ensure the sustainability in the smart kitchen environment.

1.6 Importance of this project:

The market for IoT appliances is increasing over time. These appliances are now cost-effective and easily accessible. With the collaboration of cloud services, IoT appliances can make humans life easy and comfortable which has so many examples around the world. For example, the smart city is one of the successful application of IoT appliances. However, there is no city in Australia which can point out as a smart city. Moreover, initiatives have been taken from the Australian government to ask the citizen for any innovative idea to contribute to smart city plan (Cities 2018). In that case, this project can be an additional source to make an effective plan for the smart city because the kitchen is one of the most significant places for people.

In everyday life, activity starts from the kitchen every morning, and this is the real scenario for every human being. People need to make at least one meal per day at the kitchen which has so many hassles such as checking all groceries and making a list if something needs to buy, hazardous events like gas leakage or forgetting to turn off any machine which can spark the fire.

This smart kitchen project is providing a solution to overcome these types of hassles in the kitchen. Firstly, our service can track the grocery consumption, generate an automatic shopping list for every week grocery shopping for the customer and it will also suggest the best prices for all items. There is also a service that can only notify the customer when any grocery item is finished instantly. Moreover, another service provides the daily water and gas consumption in the kitchen. Most importantly, a service can send an email if any leakage found in the gas stove or gas cylinder. In addition, there is also a service that can send mail if the temperature of the kitchen is high than the threshold (average temperature) which means anyone forgets to turn off stove, oven or microwave oven or any machine that increases the temperature which can spark the fire. Lastly, smart kitchen has a service that can keep track of the daily water and gas usage. The importance of this project can be categorised that includes:

i. Helping user to track their daily grocery consumption
ii. Reducing the hassle of making grocery shopping list
iii. Providing best price information for grocery shopping
iv. Notifying customer by email to shop urgently if any item finished instantly
v. Assisting user to track the daily energy (water, gas) consumption
vi. Notifying customer by email if any leakage found in the gas stove or gas cylinder and if the kitchen temperature is high.

1.7 Requirements:
At present, our project has three requirements which are comfort, safety and resources (Gas & Water) consumption. We developed IoT applications to fulfil these requirements which are discussed below:

1.7.1 Comfort:
This project is designed mainly to improve the system that can increase the level of human comfort. This smart kitchen project is mainly designing to assist humanity to reduce some tension in the kitchen. For example, almost all household need to go for the grocery shopping at least once per week, and the main stressing part is checking all groceries frequently to make the shopping list, and mistakenly sometimes people miss things to add in the list. To solve this type of problem our project is offering services that are assisting users to track their daily grocery consumption so that user can observe the grocery consumption (how much left) anytime at our smart kitchen website by their unique customer ID. Moreover, our service will send an email consisting a shopping list every week. As a result, there is no chance to miss out anything and user don't need to take the hassle of making grocery shopping list by themselves. Most importantly, they can decide what needs to buy by accessing the smart kitchen website when they are in the grocery store. In addition, the smart kitchen project is also providing the best price information for all groceries so that they can save money.

For the customer satisfaction, this smart kitchen project has another service that will also notify the customer if any grocery item finishes instantly. For example, customer checked on the website there are enough eggs, so I don't need to buy, but suddenly any member of the household used all the eggs, so he will get a notification email that will ask him to buy eggs.

1.7.2 Safety:
The smart kitchen project will ensure users safety at kitchen smartly. Many kitchen related occurrence (such as gas leakage which sparks the fire or keeping the stove on by mistake) can make human life unsafe so that our service has been designed to solve these issues. The user who subscribes our service can get notification instantly if any leakage found in the gas stove or gas cylinder.

Most of the kitchen has smoke detector in Australia. However, smoke will be detected after getting fire. For some kitchen, fire alarm sends the call to the firefighter team automatically; sometimes false alarm calls the firefighter, and household needs to pay the penalty for this. In that case, our service will send the prevention alert before getting the fire to save the household from any unnecessary hassle. For many reasons (such as someone leaves the oven or stove turn on), if the temperature of the kitchen is high than the average user will receive instant mail that something went wrong.

1.7.3 Resources (Gas & Water) Consumption:
This project will cover the sustainability maintaining part as well. Service has been designed to track the concern use of gas and water in the kitchen that can assist the sustainability management mechanism in Australia. User can easily track their daily gas and water consumption on the smart kitchen website by using their unique customer ID.

1.8 Structure of the rest of the report:
This project report will introduce the explanation and the architecture of smart kitchen services (user-centric), analyse cloud services (used) with the explanation of their effectiveness and the whole system. It will also explain the cost and resource planning of cloud services (used). This report will also include individual reflect of group members.

2. Project Methodology and Product Analysis
2.1 Project solution:
The smart kitchen provides services to increase the human comfort and safety and additionally it will cover the sustainability-maintained services. Mainly IoT devices are collecting the data, and it is uploading into the cloud. With the collaboration of some cloud services, all data is processed, and the user is receiving smart kitchen services through the HTTP request into the hosted website which is uploaded into the cloud.

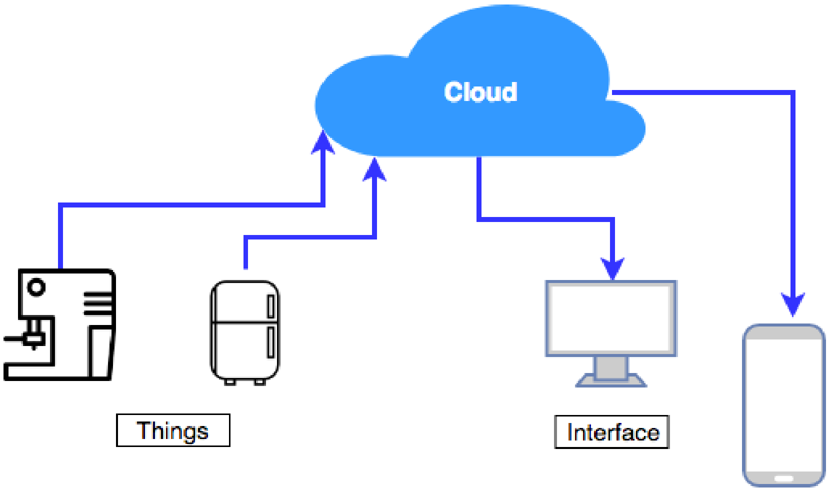

Figure 1: Project solution

2.1.1 Hosting website:

Primarily this project is designed such a way so that user can consume all the services by using website. Smart kitchen website will provide all the necessary information about all services. At first, User is required to subscribe to the smart kitchen services through the hosting website. Moreover, the user can keep track most of the services through the smart kitchen website and receives all notifications through email.

2.1.2 IoT sensors:

IoT sensors will be placed to provide all the mentioned services. There is egg sensor that can count egg, and it can sense if the tray is empty or full. The sensor can measure the milk. The weight sensor is placed in the specific place in the refrigerator, and user needs to place their meat, fish, vegetables and fruits in the specifically mentioned segment so that these sensors can send the specific data to cloud and cloud service can process the data. Moreover, gas leakage sensor and temperature sensor are also placed to ensure consumer's safety in the kitchen.

Table 1: IoT sensors

 IoT sensor: Coffee sensor	 IoT sensor: Egg detector	 IoT sensor: Weight detector
 IoT sensor: Temperature sensor		 IoT sensor: Gas leakage sensor

2.1.3 Cloud Services

IoT services will generate the data and it will be uploaded into the cloud so the main tricky part is to collaborate all necessary loosely coupled cloud services to make the best features. All public cloud providers such as Google, Amazon, Microsoft are providing platform to access all cloud services to make any kind of robust applications. We aimed to choose services from Amazon Web Services (AWS).

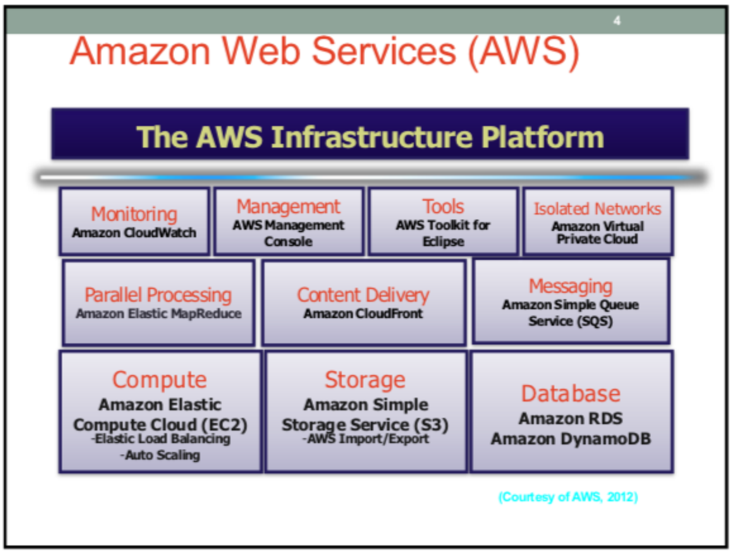

Figure 2: AWS Services (Dr. Bahman Javadi [University of Western Sydney] 2018, 301042 lecture 8, 20 October)

2.2 Considering factors while making decision:

The main considering factors for this project will be how to maintain all the sensors, the power supply of those sensors and security issues. However, this project is conducting with limited scopes, so we exclude these parts. For the service part, we need to consider the cost and budget for the whole project because if we want we can use any cloud services for the same project solution, but it needs to be less expensive. To provide efficient services, customer needs to be notified how to use those sensors and how to place all groceries according to the weight scale. Moreover, there is a sensor that will notify the user by instant email, if the temperature of the kitchen is high than the average, in that case, customer need to be well notified to turn off the sensor while cooking.

3. Detailed description of the solution (Explain architecture)

3.1 Our choice: Select Cloud service provider

In this project, we have used Amazon Web Service (AWS) for deploying the Smart Kitchen architecture. AWS is one of leading cloud service provider, it offers the wide selection of services, and for SME (Small and Medium Enterprises), it provides trusted cloud-based solution based on business requirements (Fortune, 2018 and Amazon Web Services, Inc., 2018). Moreover, there are lots of projects already available in AWS, which help to do the new project easily. Besides with this, AWS offers 12 months' free tier account option that help developer to create and test their architecture. AWS also provides to educate account which not only provides credit for students or teachers but also give free AWS online learning platform. In our project, we got AWS Educate account from the university.

3.2 System architecture

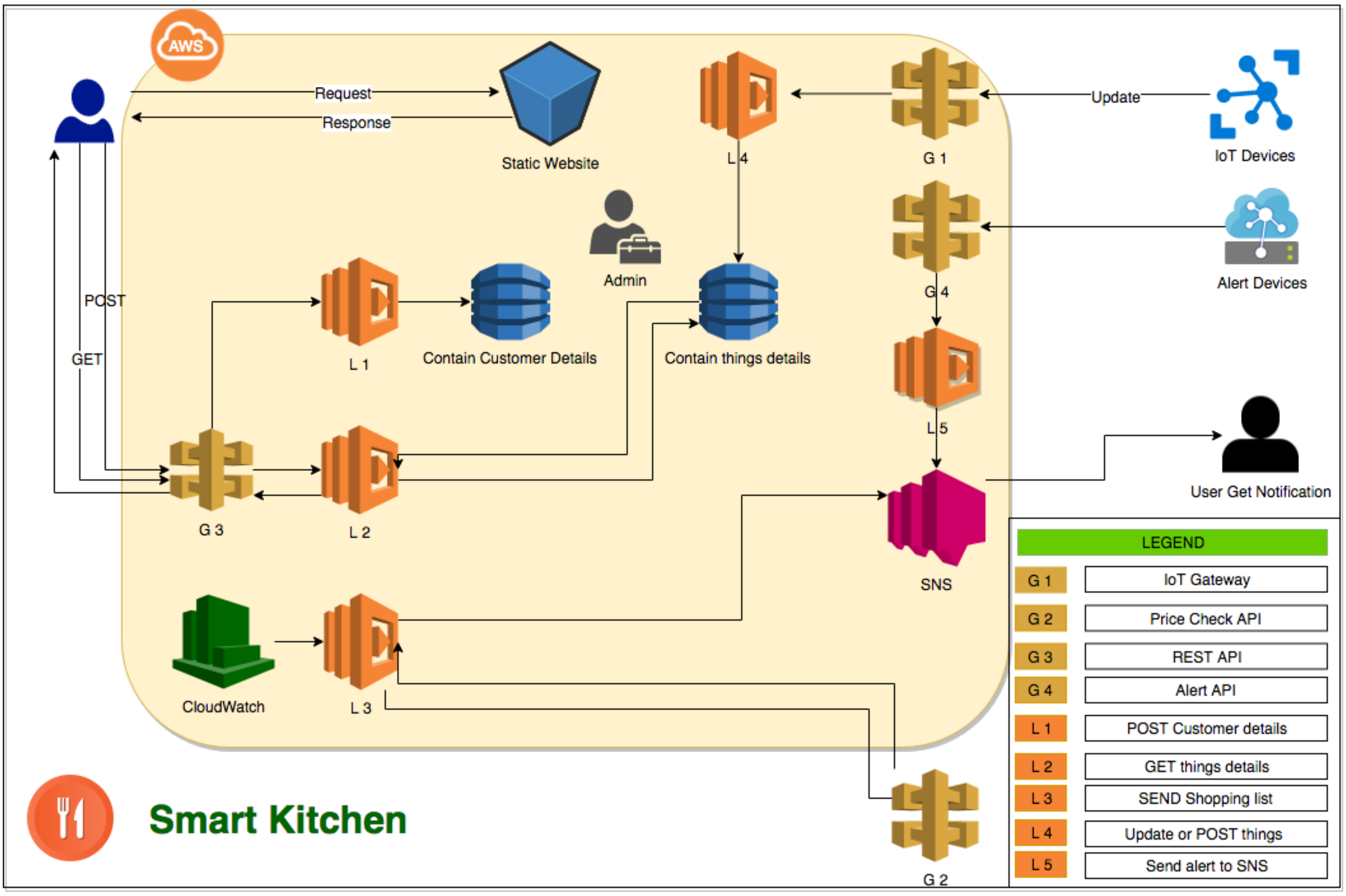

Figure 3: Smart kitchen AWS architecture.

User Registration and static website:

- The static site is hosted in S3; the user can send a request about the content to S3 and response back the content from S3.
- The user can register to the services by sending POST request through REST API gateway (G3), and it will trigger lambda function (L1). The customer details like (ID, email, First Name, Last Name, Mobile etc.) will save it to the DynamoDB.
- Admin will use this email ID to send SNS subscription request to the user.

Sending IoT devices data and GET request:

- IoT devices will transmit data through IoT gateway (G1), and it will trigger lambda function (L4). The data will be saved to DynamoDB.
- The user can access to this data by sending GET request through REST API (G3), and it will trigger lambda function (L2). The lambda function (L2) will access the DynamoDB data and response back to the user.

Get alerts for the emergency situation:

- The user will get an email notification if the temperature of the kitchen appliances or/ and gas leakage happen. IoT alert devices will send an alert through alert API (G4), and API will trigger lambda L5 and send alert to SNS. The user will get email notification through SNS.

Shopping list:

- Every week CloudWatch will trigger lambda (L3), which will get a shopping list from DynamoDB and send it to the user through SNS.
- Lambda (L3) also check the price of the product from price check API (G2) and send email to the user.

4. Cloud Technologies Used

4.1 AWS Lambda:

It is one of the compute services in AWS which has less worry about the server. It works as function as a service which allow developer to run any code without managing the server. The user codes are executed only when the user requires, and it scales automatically. The service is based on as per usage so that the user is not charged unless the code is not running. It provides the user flexibility to run code for any virtual application so for this project we chose AWS lambda to minimise the resource management hassle.

Our smart kitchen project is providing all event-driven services, so Lambda AWS lambda service is being used. Most of the requirements of this project are GET, PUT and Update request.

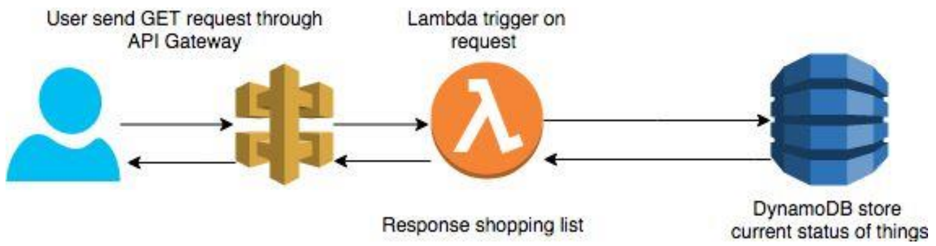

Figure 4: Lambda working mechanism

4.2 API gateway:

Developer can create API's that can access various web services and also the data that is stored in the Aws Cloud by the use of AOI gateway in AWS. API gateway is designed to help the users in delivering secure web, and mobile application backends.

In this project API gateway is used to receive sensors update; GET and response user request. Similar like lambda, AWS API charges only when it will be used. It is also fully managed service and it easier to "create, publish, maintain, monitor" API. Eventually it creates REST API that will help user to call lambda from anywhere ("Amazon API Gateway", 2018).

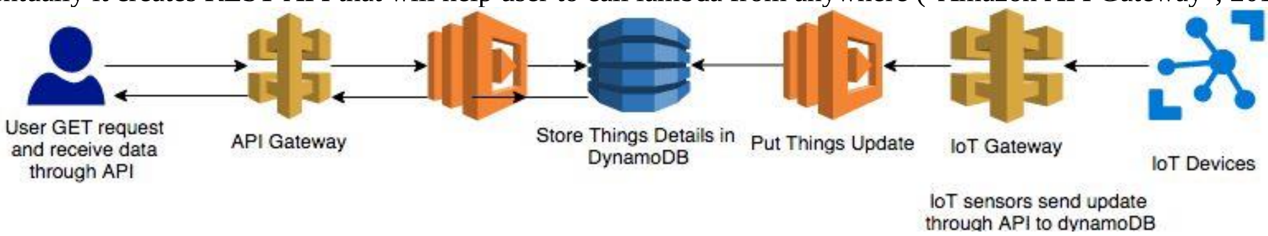

Figure 5: API Gateway working mechanism

4.3 AWS S3

Amazon S3 (Simple storage Service) is object storage that is built to store and access data from different sources such as website, mobile applications as well as the smart, physical devices (IoT sensors). Amazon S3 can store the significant amount of data for different applications and guarantees 99.99% durability. It provides users with flexibility in managing the data for cost optimisation, efficiency and compliance. In this project we used AWS S3 to host the smart kitchen static website.

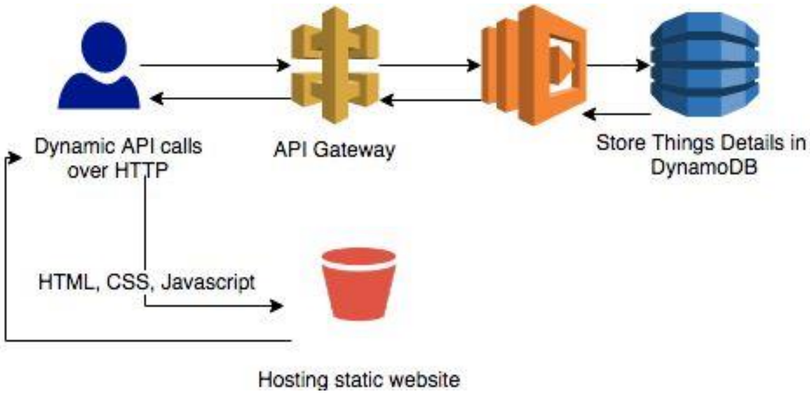

Figure 6: S3 mechanism (How to Build a Serverless Web Application with AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB, and Amazon Cognito | AWS. 2018)

4.4 AWS DynamoDB

Dynamo DB is a non-relational database designed to deliver high-quality, reliable performance at any scale. It is a fully managed server-less database that continuously monitors and backs up user's data for protection. The user's data is fully encrypted, and the Service level encryption guarantees high reliability of data.

In this project we used DynamoDB because it is proper if the application requires more read and write data. Moreover, DynamoDB provides low latency for IoT data and it stores unstructured data, so it is flexible to add more data frequently.

4.5 CloudWatch and SNS

AWS CloudWatch is a monitoring service that can play a role as controller. In this project, we used CloudWatch for auto-scheduling to send the shopping list to the consumer so that customer can receive shopping list within 7 days.

AWS simple notification service (SNS) is used to carry out messages to subscribers of the service on a large scale. It gives the user liberty to keep the sensitive data secure by setting topic policies which can help the user to restrict as to who can subscribe or publish to a topic. Smart kitchen project used SNS to send any notification mail to users.

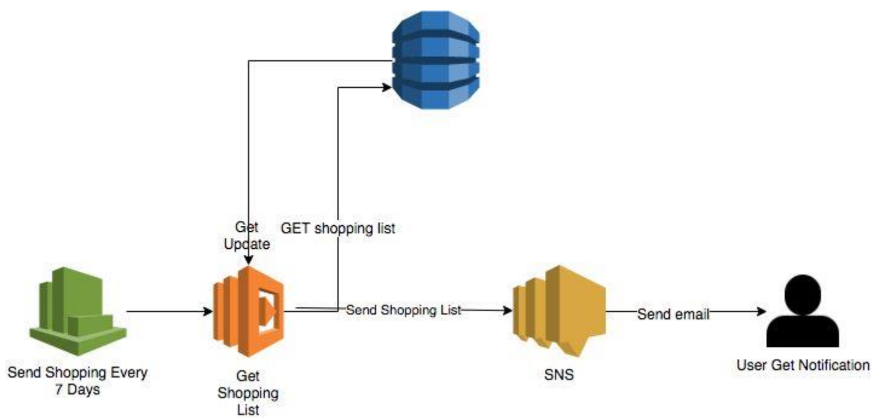

Figure 7: CloudWatch and SNS mechanism

5. Cost Analysis for Cloud Resources

This section describes the cost analysis. For testing purpose, cost analysis is done based on 10000 households, and each household has 10 things hence 10 IoT sensors connected to detect milk level, eggs, meat etc. Suppose one person in each household has access the main interface (Web or mobile). On average user check their mobile phone 86 times a day, so it is considered that user will send GET request 86 times a day (Wolfe, 2018). The ideal scenarios that, is used throughout the calculation are:

Table 2: Necessary estimation for the whole project cost calculation.

Total number of household	**10000**
Each household have	10 sensors
Total number of IoT sensors	10 * 10000 = 100000
User send GET request	86 times a day

5.1 Lambda

Aws Lambda is used in this project instead of any other services because execution of lambda is comparatively cheaper than EC2 and its good for short running request ("What Is AWS Lambda? - AWS Lambda", 2018). The details calculation of lambda function is described below

Suppose the provider allocated 128MB of memory for execution of getting Update lambda function, 10000 households if use request GET 86 times a day so total: 86*10000=860000. So, in one month it would be 25800000 or 26 million requests. If it ran for 1 second each time, charges would be calculated as follows ("AWS Lambda – Pricing", 2018):

Table 3: Estimation for the Lambda calculation

Monthly compute price per GB (MC)	$0.00001667
Running time (R)	1 second
Monthly request price (MP)	$0.20 per 1 million requests

Table 4: Lambda calculation table.

Lambda Function	Execution request per months (E)	Total Compute in seconds (C)=(R*E)	Total Compute GB-s (TC)=C*Allocated Memory (GB)/1024	Monthly compute charges = TC*MC	Monthly request charges (MRC) =E*MP	Total Cost =MRC+ Monthly compute charges
Get update	26000000	26000000	3250000	$ 54.18	=26 M * 0.2/M = $5.2	=54.18+5.2 =$59.38
Put things update	12000000	12000000	1500000	$25.005	=1.5M*0.2/M =$0.3	=25.005+0.3 =$25.305
Get Shopping list	40000	40000	5000	$0.08335	=0.004M * 0.2/M =$0.008	=0.08335+0.0008 =$0.08415
Total						**$84.77**

5.2 API Gateway

Similar like GET lambda function, if API gateway receives 86 requests from one household, so for 10000 households: 86*10000=860000 request will be received a day. Like that if 10 sensors send update 4 times a day than total API call will be: 10000*10*4*30= 12 million. Therefore, for one month it will receive: 25800000 or 26 million requests. If each time, it sends 1kb data than calculation will be like following ("Amazon API Gateway – Pricing", 2018):

Table 5: Estimation for the API Gateway calculation

AWS API call charge (AC)	$3.5/ million
Data transfer charge (DTC)	$0.09/GB

Table 6: API gateway calculation table.

AWS API	Total API call per months (TAC)	**Total API call charge (TACC) (AC*TAC)**	**Data transfer (DT) = 1KB*TAC**	**Total API gateway data transfer charge (TAGD) = DT*DTC**	**Total API charge= TAGD+TACC**
GET request	26 million	26*3.5=$91	26 million/KB=26 GB	26*0.09=$2.34	91+2.34=$93.34
Sensors update request	12 million	12*3.5=$42	12 million/KB=12 GB	12*0.09=$1.08	42+1.08=$43.08
Total					**$136.42**

5.3 AWS S3

The static website is being hosted in AWS S3, the reason behind this is, the user only uses the website to see the contents like smart kitchen services and whenever to want to see the status of the sensors, they can send GET request to lambda through API. Moreover, S3 is comparatively cheaper than other AWS storage like EBS, EFS (Kovacs, 2018). The whole cost calculation for S3 depends on the storage saved in it.

Table 7: Estimation for the AWS S3 calculation

Total website (Smart Kitchen) file size	**9 MB**
Total GET request	26 million (see section 5.2)

5.4 DynamoDB:

AWS DynamoDB provides fully managed non-relational database service; it's proper the application that requires low latency like IoT and its store unstructured data therefore easy to add more data ("Amazon DynamoDB - Overview", 2018).

Table 8: Estimation for the DynamoDB calculation

Total database size for 10000 household	**1KB*100000/ 0.01GB**
Number of Read request per month	26 million (see section 5.2)
Write request per month	12 million

5.5 CloudWatch and SNS

Cloud watch and SNS are used only once a week. Cloud watch is configured to send request lambda in once in week and lambda will send shopping list through SNS.

Number of CloudWatch and SNS request per month is: 4 * 10000 = 40000

AWS pricing calculator is used to calculate following services in AWS

Table 9: AWS S3, DynamoDB, CloudWatch, SNS calculation *(Source: Calculator.s3.amazonaws.com, 2018)*.

Service Type	**Components**	**Component Price**	**Service Price**
Amazon S3 Service (Asia Pacific (Sydney))			$11.43
	S3 Standard Storage:	$0	
	S3 Standard GET Requests:	$11.43	
Amazon DynamoDB Service (Asia Pacific (Sydney))			$7.05
	Provisioned Throughput Capacity:	$1.63	

	Indexed Data Storage:	$0	
	DynamoDB Streams:	$5.42	
	On-demand backup:	$0	
	Continuous backup (PITR):	$0	
	Restoring a table:	$0	
	Global Tables:	$0	
Amazon CloudWatch Service (Asia Pacific (Sydney))			$0
	Dashboard:	$0	
Amazon SNS Service (Asia Pacific (Sydney))			$0.78
	Requests:	$0	
	Notifications:	$0.78	
AWS Data Transfer Out			$0
	Asia Pacific (Sydney) Region:	$0	
AWS Support (Basic)			$0
	Support for all AWS services:	$0	
Total			**$19.26**

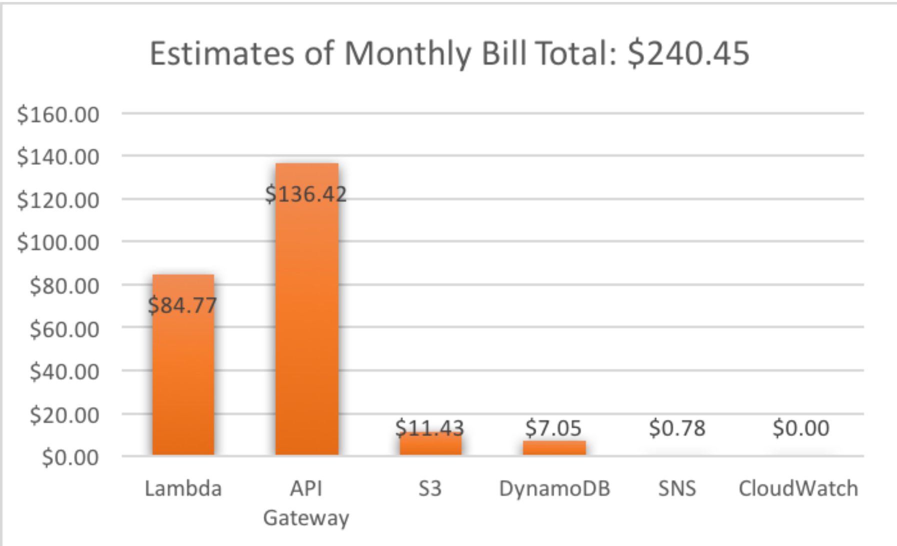

Figure 8: Breakdown of total monthly bill of various AWS services

It is evident from the graph that API Gateway cost significant amount around $136.42, following by this lambda cost around $84.77 per month. CloudWatch would not make any charge as it uses minimal times like once a week. If 10000 households use this service, per household cost will be $0.024 only per month.

Table 10: Monthly and yearly total cost

Smart Kitchen whole project cost	Monthly	Yearly
Total service cost for 10000 households	$240.45	$2885.4

6. Conclusion, Future Work, Plans for Extension

This smart kitchen project is providing solutions to overcome the kitchen-centric hassles. The main scope of this project is to increase the level of human comfort in the kitchen by generating weekly shopping a list. Another scope is providing safety features by notifying the user about any leakage in the gas stove or gas cylinder and the kitchen temperature if it is high. Lastly, it will also ensure the sustainability in the smart kitchen environment by notifying the user about the daily resources (gas and water) consumptions. The basic concept of this project is that IoT sensors are generating data and uploading the data into the cloud where the necessary collaboration of cloud services is processing all data and showcasing according to the user demand. We used AWS to design the architecture of Smart Kitchen project because of its flexibility and feasibility services for developers. From AWS services this project used six services (AWS Lambda, API gateway, S3, DynamoDB, CloudWatch and SNS) after considering the efficiency of all services. Our static smart kitchen website is hosted into S3 so the user can subscribe to all smart services by POST request and the user can observe the grocery consumption (how much left) anytime by GET request. Multiple Lambda functions are triggered to read the sensors data from DynamoDB and send back to the user. Similarly, IoT devices are also sending data through API gateway and triggered Lambda which is saved into DynamoDB. Lastly, the user is getting the shopping list and best price information through SNS in their mail. All these services we used they are truly cost-effective for this project. We estimated that if 10000 households subscribed this project, then the total monthly cost for cloud services will be $240.45 and if we run this project for a year the cost will be $2885.4.

Currently, this project address human comfort, monitoring resources consumption and user safety. Besides, with this, it provides only the web interface to the user to monitor their devices. In future, more features will be added like energy consumption, daily food consumption, connect all kitchen appliances like the bean, dishwasher, Micro-oven and so on. A user will be able to access the interface through mobile and tablet. More features will be added to the interface like comparing current weekly consumption with the previous week, see total monthly consumption and so on. The cloud architecture will also be updated in future. For example, AWS SQS (Simple Queuing Services) will be added to IoT gateway that will accept the high volume of IoT data and put in the queue. It will help provide the scalable solution to hand a large amount of data by dynamically increasing read throughput.
Furthermore, AWS CloudFront will be used that help user to access their interface (Web or Mobile) globally securely. However, at present user only get the email notification with simple plain text. In future, the user will get the mobile notification, as well as AWS SES (Simple Email Service), will be used to send the nicely formatted shopping list to the user.

7. Individual Reflections

Arman (18897816)
This unit has helped to get the depth of understanding about cloud computing including not limited to virtualisation, API programming, Map-reduce programming, Server-less programming and so on. I came to know various public cloud services provider like Amazon, Google, Microsoft and so on. Moreover, Amazon is one of the leading cloud services, and it provides free tier account and educates account. Therefore, we had to do the project by using Amazon Web Service with a group of three people. This has helped me not only use cloud service to do the project but also increased my team work skill. At the beginning of the project, we formed a problem description and discussed it with our unit coordinator. After reviewing with the problem description, we found that Smart Kitchen is one of the subdomains of Smart City and IoT is the underlying factor to make the kitchen Smart. We started to use AWS to do our project, and we used several AWS services like lambda, DynamoDB, CloudWatch and so on. I became familiar with several AWS services and got the hand on experience in doing the project. Although we did not use real IoT devices, we sent the data externally through API gateway. We divided our work equally where my task was to work mainly on lambda function. We discussed with each other about which services will be feasible regarding cost and performance. I found that Server-less service like lambda help to run them without providing the whole server and reduce the cost and hassle to manage the server.

Lastly, I got a real-life experience to make a cloud-based project, apart from this I have learned how to brainstorm of specific problem and forming and planning project.

Nusrat (19060343)
This unit and this whole project have given me the theoretical and practical knowledge of using cloud services (IaaS, PaaS, SaaS,) to build any robust application based on the public cloud platform (AWS). I have introduced with renowned cloud service providers like Microsoft, Amazon, Google and so on. I have learned the basic concept of cloud such as how the cloud is providing IT utility as pay as you go mechanism. I have learnt about so many AWS services in the lecture and tutorial that assist me to run this project on AWS platform. This whole project has boosted up my knowledge about the cloud-based application. We have done the Smart kitchen project which was not only an assessment for me it gave me the flavour of working in the real field. I worked with other two members in our group, so we had to discuss n divide our work, and sometimes we had to negotiate each other which was fun. At first, when we started, it was so difficult to figure out the efficient services for this particular project, but AWS has all the necessary description for their services. At first, we started designing our smart kitchen services into the AWS educate account. However, AWS account has some limitation of accessing all services, so we ended up using the free tier account to design our project.

We have used real cloud services such as Lambda, S3, DynamoDB, CloudWatch, API gateway, SNS for this project. I assumed we had used these services successfully to receive the best outcome from the IoT applications. This topic can be an essential initiative of launching a smart city in Australia.

8. References

Amazon API Gateway. (2018). Retrieved from https://aws.amazon.com/api-gateway/

Amazon DynamoDB - Overview. (2018). Retrieved from https://aws.amazon.com/dynamodb/

Amazon Web Services, Inc. (2018). AWS for SMEs. [online] Available at: https://aws.amazon.com/switzerland/smes/ [Accessed 29 Oct. 2018].

AWS Lambda – Pricing. (2018). Retrieved from https://aws.amazon.com/lambda/pricing/

Amazon API Gateway – Pricing. (2018). Retrieved from https://aws.amazon.com/api-gateway/pricing/

Calculator.s3.amazonaws.com. (2018). Amazon Web Services Simple Monthly Calculator. [online] Available at: https://calculator.s3.amazonaws.com/index.html [Accessed 28 Oct. 2018].

Cities. (2018). Cities. [online] Available at: https://cities.infrastructure.gov.au [Accessed 17 Sep. 2018].

Fortune. (2018). http://fortune.com. [online] Available at: http://fortune.com/2017/06/15/gartner-cloud-rankings/ [Accessed 29 Oct. 2018].

How to Build a Serverless Web Application with AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB, and Amazon Cognito | AWS. (2018). Retrieved from https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Khan, R., Khan, S.U., Zaheer, R. and Khan, S., 2012, December. Future internet: the internet of things architecture, possible applications and key challenges. In Frontiers of Information Technology (FIT), 2012 10th International Conference on (pp. 257-260). IEEE.

Kovacs, G. (2018). Amazon EBS, EFS or S3: Comparing Pricing, Performance and More. Retrieved from https://cloud.netapp.com/blog/ebs-efs-amazons3-best-cloud-storage-system

What Is AWS Lambda? - AWS Lambda. (2018). Retrieved from https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

Wolfe, A. (2018). Guess how often you use your phone every day. Retrieved from https://www.journalofaccountancy.com/newsletters/2018/apr/how-often-use-phone-every-day.html