

# **INTERNSHIP REPORT**

*A report submitted in partial fulfillment of the requirements for the Award of Degree  
of*

## **BACHELOR OF TECHNOLOGY**

**In**

## **COMPUTER SCIENCE AND ENGINEERING**

**By**

**STUDENT NAME: Ankit Kumar**

**ENROLLMENT NO: 0191CS201024**

**Under Supervision of  
Mr. Amar Nayak**

**(Duration: 20<sup>th</sup> June, 2022 to 13<sup>th</sup> August, 2022)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**TECHNOCRATS INSTITUTE OF TECHNOLOGY (EXCELLENCE) BHOPAL**

**SESSION: 2022-23**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
**TECHNOCRATS INSTITUTE OF TECHNOLOGY (EXCELLENCE) BHOPAL**



***CERTIFICATE***

This is to certify that the “AWS Cloud Practitioner Certification Internship Report” submitted by **ANKIT KUMAR, 0191CS201024** work done by him and submitted during 2022 – 2023 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**, at Ramraj Technology Solutions Private Limited.

Department Internship Coordinator

Prof. Amar Nayak

Assistant professor, CSE

Prof. Rajesh Boghey

Head of the Department (CSE)

**CERTIFICATE PASTE HERE**

# ACKNOWLEDGEMENT

First I would like to thank **Mr. Amar Nayak of Ramraj Technology Solutions Private Limited, Bhopal** for giving me the opportunity to do an internship within the organization.

I also would like all the people that worked along with me **Ramraj Technology Solutions Private Limited, Bhopal** with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Director **Prof. (Dr.) K.K DWIVEDI** for the facilities provided to accomplish this internship.

I would like to thank my Head of the Department **Prof. Rajesh Boghey** for his constructive criticism throughout my internship.

I would like to thank **Prof Amar Nayak** internship coordinator Department of CSE for their support and advices to get and complete internship in above said organization.

I am extremely great full to my department staff members and friends who helped me in successful completion of this internship.

**Student Name: ANKIT KUMAR**

**Enrollment Number: 0191CS201024**

# AWS Cloud Practitioner Certification Internship Report

## Abstract

Computers have turned into a vital piece of life. We require computers everywhere, be it for work, research or in any such field. As the utilization of computers in our everyday life expands, the computing resources that we need also go up. For companies like Google and Microsoft,

Cloud computing is a paradigm shift in which computing is moved away from personal computers and even the individual enterprise application server to a 'cloud' of computers. Which can give the distinctive registering assets of their customers? Clients of this framework require just be worried about the computing resources administration being requested. The fundamental points of interest of how it is accomplished are avoided the client. The data and the services provided reside in massively scalable data centers and can be ubiquitously accessed from any connected device all over the world. Google, Microsoft, and Amazon, Alibaba Rackspace has started providing cloud computing services. Amazon is the pioneer in this field.

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

Cloud computing is a growing technology which could change traditional IT systems. Cloud computing makes it feasible for an organization IT to be more flexible, save costs and process information and data faster than with traditional IT. The problem though lies in the riskiness of this new technology.

Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. Cloud computing is attractive to business owners as it eliminates the requirement for users to plan ahead for provisioning, and allows enterprises to start from the small and increase resources only when there is a rise in service demand. However, despite the fact that cloud computing offers huge opportunities to the IT industry, the development of cloud computing technology is currently in its infancy, with many issues still to be addressed.

Cloud computing has gained a lot of publicity in the current world of IT. Cloud computing is said to be the next big thing in the computer world after the internet. Cloud computing is the use of the Internet for the tasks performed on the computer and it is visualized as the next generation architecture of IT [1].

## 1.2 Motivation

I was highly motivated science I got my first class on cloud computing at BITM. Course coordinator was an expert cloud professional. He gave us a good overview and told us the future of cloud computing. The most important things are:

**Scalability** – Cloud computing is highly scalable. Use scalability we can scale up and down our cloud resources. A cloud-based IT infrastructure is more versatile – notably in terms of scalability – that is local, intranet-based infrastructure.

**Reliability** – Cloud computing service providers provide stable and reliable resources. They provide up to 99.99 % uptime. They make multiple copies of our resources and our data and spread them multiple regions.

**Virtualization** – Because cloud-based IT infrastructure can be virtualized and geographically dislocated, startups are freed from having to consider the physical location of its IT infrastructure and data centers in business operations decisions.

**Affordability** – Under traditional infrastructures, startups may not receive – or have the financial wherewithal to purchase – certain features that are often offered to cloud computing customers at substantial discounts. How do these benefits pass on to startups and other small companies? Because the marginal cost to the cloud computing provider of many features (such as enhanced security) may be very low (or even negligible), otherwise unaffordable services may be offered for free to startups using cloud computing options.

## 1.3 Internship Objectives

The main objective of this internship is getting basic knowledge about cloud computing and load balancing. But the main part is designing a highly available, cost effective, fault tolerant, scalable system. That's why we need to Understand and use:

- core services of Amazon Web Services (AWS)
- Identity & Access Management (IAM)
- Virtual Private Cloud (VPC)
- Simple Storage Service (S3)

- Elastic Cloud Compute (EC2)
- RDS/DynamoDB (databases)
- Simple Notification Service (SNS)
- CloudWatch (monitoring)

## 1.4 Introduction to the Company

Ramraj Technology Solutions Private Limited is a software development company with most of the modern technology services including Cloud Computing, Mobile Application Development, Web Application Development, Web Development and Digital Marketing.

## 1.5 Report Layout

**Chapter - 1** described of introduction, motivation, objectives and expected the outcome of the internship.

**Chapter - 2** covered the methodology of my internship and company enterprise.

**Chapter - 3** showed a UML diagram and practical work of various cloud services and my daily task, activities events etc. How load balancing work, load balancing configuration and output of real life work.

**Chapter - 4** Skills that I developed. Which skills are more important and which was more fun to me is explained.

**Chapter - 5** Discussion and Conclusion added in this section. What is my future plan and what is about career is explained.

# CHAPTER 2: ORGANIZATION

## 2.1 Introduction

I have taken my internship at ramraj Technology Solutions Private Limited. It is involved in Software publishing, consultancy and supply [Software publishing includes production, supply and documentation of ready-made (non-customized) software, operating systems software, business & other applications software, computer games software for all platforms. Consultancy includes providing the best solution in the form of custom software after analyzing the user's needs and problems. Custom software also includes made-to-order software based on orders from specific users. Also, included are writing of software of any kind following directives of the users; software maintenance, web-page design].

## 2.2 Product and Market Situation

### Web Solutions

- Domain Registration
- Web Hosting
- Website Design
- Word Press Solutions

### Software Development

- Web Application
- API Development
- CRM Application
- Business Automation
- Batch Processing Tasks
- Monitoring & Security
- Big data Analysis

### Cloud Computing

- Server Deployment
- Monitoring & Reporting
- Load Balancing & Scale
- CDN Solution
- DevOps
- Cloud VPN
- Cloud DNS
- BigData Processing
- Data Warehouse



# CHAPTER 3: TASK PROJECTS AND ACTIVITIES

## 3.1 Daily Task and Activities

- Regularly Monitoring and capturing system log files.
- Regularly Monitoring and capturing data of server CPU, memory usage
- Regularly monitoring File systems space.
- Regularly monitoring capturing Network usage.
- Taking backup of Important file systems
- Maintaining documentation.
- Updating with new technologies.

## 3.3 Project Task and Activities

In this section, I am going to explain my working experience. How I made a highly available, cost effective, fault tolerant, scalable cloud system. Which is very much efficient for any type of organization? Work largest organization is now going to cloud and use their infrastructure and build their own system.

### 3.3.1 Architecture of cloud service UML diagram

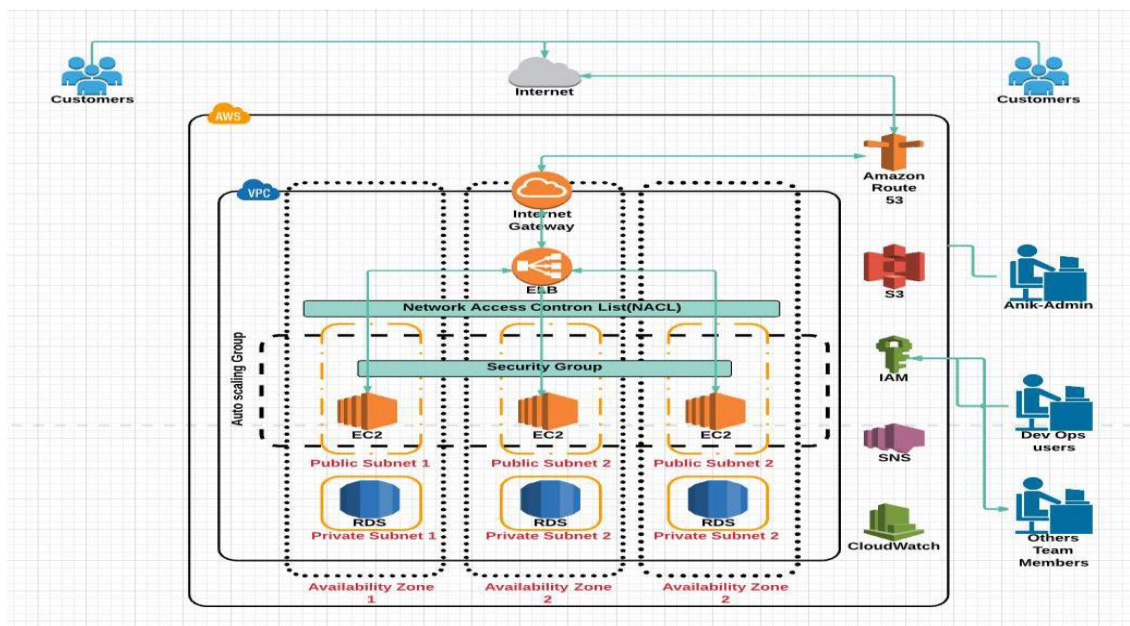


Figure 3.3.1: UML diagram of my system

This is the main part of my internship work. I created a highly available, cost effective, fault tolerant, scalable cloud system shown at figure 3.3.1. Which is very efficient, cost-effective and user-friendly? I will describe briefly how I build this system and how it works.

### 3.3.2 Elastic Compute Cloud (EC2)

#### Overview

EC2 stands for Elastic Compute Cloud. EC2 is a virtual machine. Where we can create and develop our own web server/web applications. We can create our instance by choosing specific Availability Zone [4]. Figure 3.3.2 showing how ec2 works.

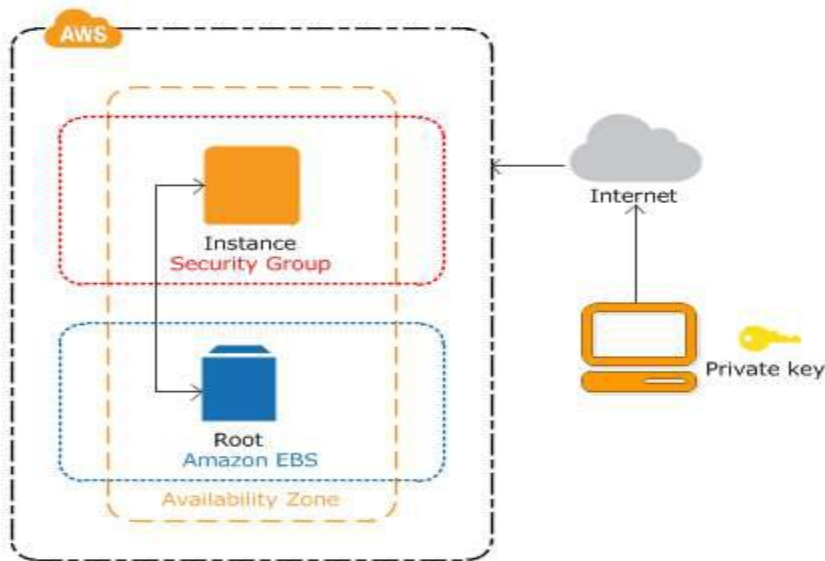


Figure 3.3.2.: How EC2 works

#### Launch an Instance

**Step1:** At first, we have to go to <https://console.AWS.amazon.com/ec2/>. Here we will see EC2 dashboard with details about EC2.

**Step2:** Select Launch Instance. Process shown at figure 3.3.2

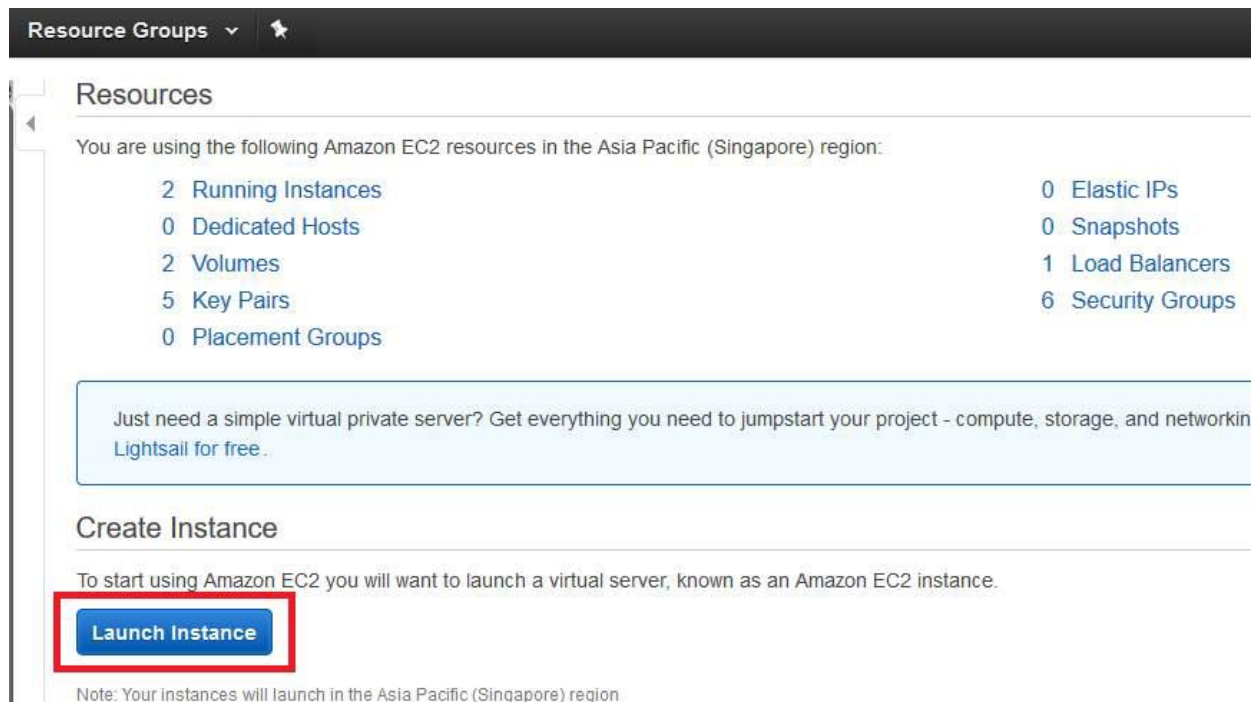


Figure 3.3.2: Launch (EC2)

**Step3:** After that, we have to select an Amazon Machine Image (AMI). Where we can select Either Linux or Ubuntu or Windows operating system. But for our own purpose, I select Ubuntu 16.04 shown at 3.3.2.

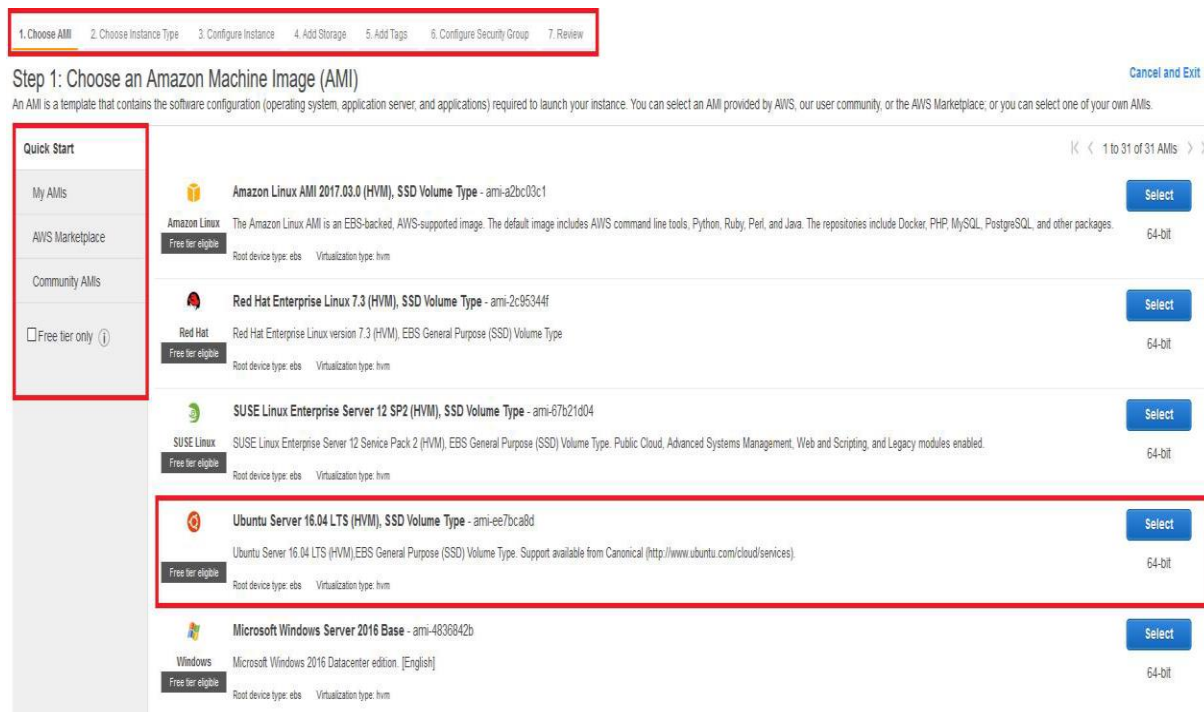


Figure 3.3.2: Select Amazon Machine Image (AMI)

**Step 4:** After that, we have to select Instance Type. I am using AWS free account that's why I have some limitation. I choose t2.micro instance which consists 1 vCPU, 1 GB Memory(EBS Elastic Block Storage) and 1 network card. Then select Configure instance.

**Step 5:** Now we will set a number of instances we need, set Virtual Private Cloud(VPC), subnet. Public IP, IAM role. Then select add storage.

**Step 6:** Here we set how many root volume we need. Here we will set 8 GB of General Purpose Storage. Then select Add Tag.

**Step 7:** In this section, we will add tag name "DaffodilEC2\_1". Then select Security Group.

**Step 8:** Here we have to create a security group for our EC2 machine named "DaffodilSG". We will add HTTP and https rules and which rules we want for our instance. Then click a Review and Launch. After checking all is ok we launch the instance then AWS tell us to create a private key, AWS will save the public key. When we will log into our instance this key will be needed? Otherwise, we won't able to log in. So this key very Important. After that, we will Launch the Instance. Now we can see our instance in the Instance section. By selecting an instance we can the details overview of that instance, process shown at figure 3.3.2.

After that, we have to log in our instance through Putty. We need to fill out Public IP, username and our Private key that we saved while we were launching our instance.

The screenshot displays the AWS Management Console interface. On the left, the navigation menu is visible, with 'INSTANCES' selected. The main content area shows a table of EC2 instances. The first instance, 'DaffodilEC2\_1', is highlighted with a red box. Below the table, the 'Description' tab is active, showing detailed information about the instance. A red box highlights the instance details, including its ID, state, type, and various configuration parameters.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
DaffodilEC2_1	i-001d9644c2432c77c	t2.micro	ap-southeast-1a	running	2/2 checks ...	No Data	ec2-54-169-148-197.ap-southeast-1.compute.amazonaws.com	54.169.148.197	-	DaffodilKey
	i-0da32d721d9b2414	t2.micro	ap-southeast-1b	running	2/2 checks ...	None	ec2-52-221-222-11.ap-southeast-1.compute.amazonaws.com	52.221.222.11	-	DaffodilKey

Instance: i-001d9644c2432c77c (DaffodilEC2\_1) Public DNS: ec2-54-169-148-197.ap-southeast-1.compute.amazonaws.com

Description		Status Checks		Monitoring		Tags	
Instance ID	i-001d9644c2432c77c	Public DNS (IPv4)	ec2-54-169-148-197.ap-southeast-1.compute.amazonaws.com				
Instance state	running	IPv4 Public IP	54.169.148.197				
Instance type	t2.micro	IPv6 IPs	-				
Elastic IPs		Private DNS	ip-172-31-38-153.ap-southeast-1.compute.internal				
Availability zone	ap-southeast-1a	Private IPs	172.31.38.153				
Security groups	DaffodilSG - view inbound rules	Secondary private IPs					
Scheduled events	No scheduled events	VPC ID	vpc-4db13899				
AMI ID	ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-20170222 (ami-9dc1706e)	Subnet ID	subnet-9d174f99				
Platform	-	Network interfaces	eth0				
IAM role	ec2	Source/dest. check	True				
Key pair name	DaffodilKey	EBS-optimized	False				
Owner	466174270526	Root device type	ebs				
Launch time	March 28, 2017 at 3:50:36 PM UTC+6 (178 hours)	Root device	/dev/sda1				
Termination protection	False	Block devices	/dev/sda1				
Lifecycle	normal						
Monitoring	basic						
Alarm status	1 of 1 in No Data						

Figure 3.3.2: EC2 Details

### 3.3.3 Simple Storage Service (S3)

S3 Stands for Simple Storage Service. S3 is an online, bulk storage service that you can access from almost any device. We can store and retrieve our data anytime from anywhere using S3 services. Figure 3.3.3 showing how s3 works [5].

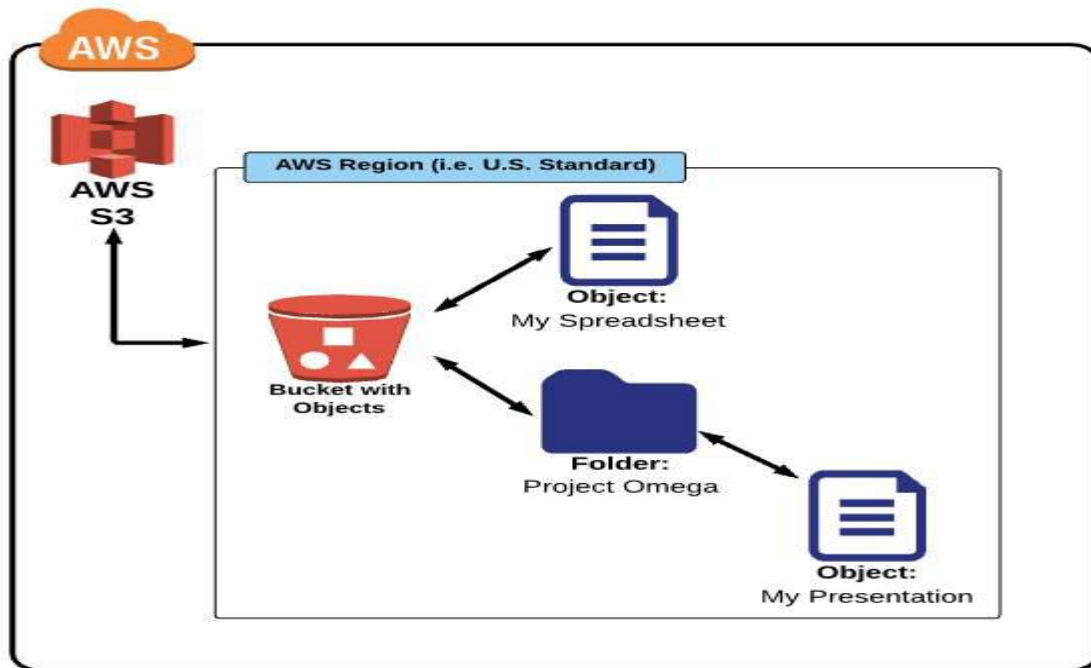
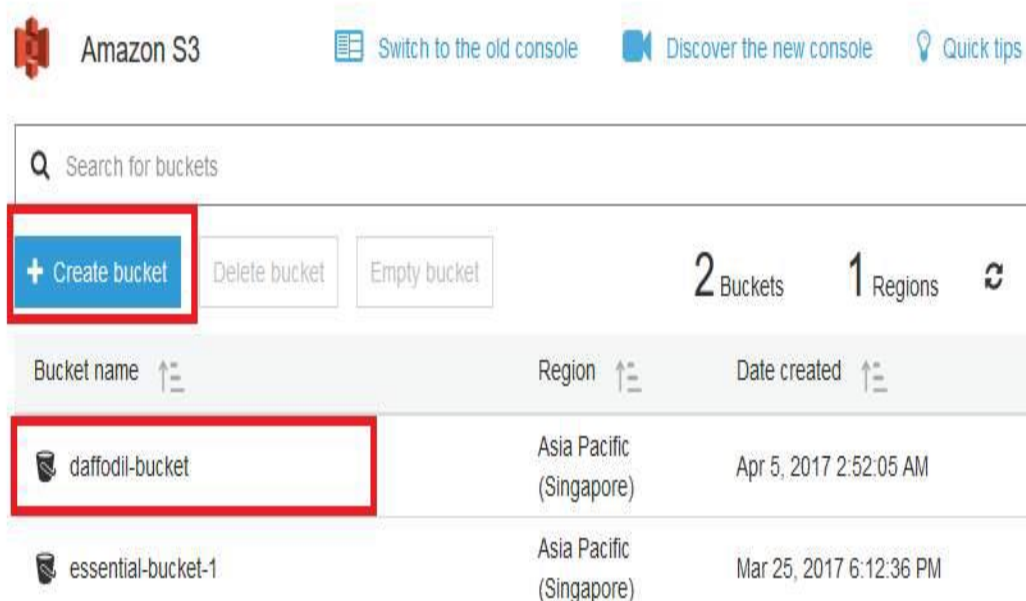


Figure 3.3.3: How S3 works

### Create S3 Bucket

**Step 1:** At first, we have to go to <https://AWS.amazon.com/s3/>. Here we will see S3 dashboard with details.

**Step: 2.** Select create a bucket. A popup window will show in front of us. See figure 3.3.3 for details overview.



**Figure 3.3.3: Create S3 bucket**

**Step 2:** Give a bucket name “daffodil-bucket”. After that, we have to select a region “Singapore”. Select next.  
**Step 3:** now we have to enable/disable some S3 properties such as Versioning, Logging, Tag etc. Select next.  
**Step 4:** In this section, we will set some permissions. Process shown at figure 3.3.3. Manage user sections we can set permission which user can do what or what can't do. We also can set the public permission. Read/write.

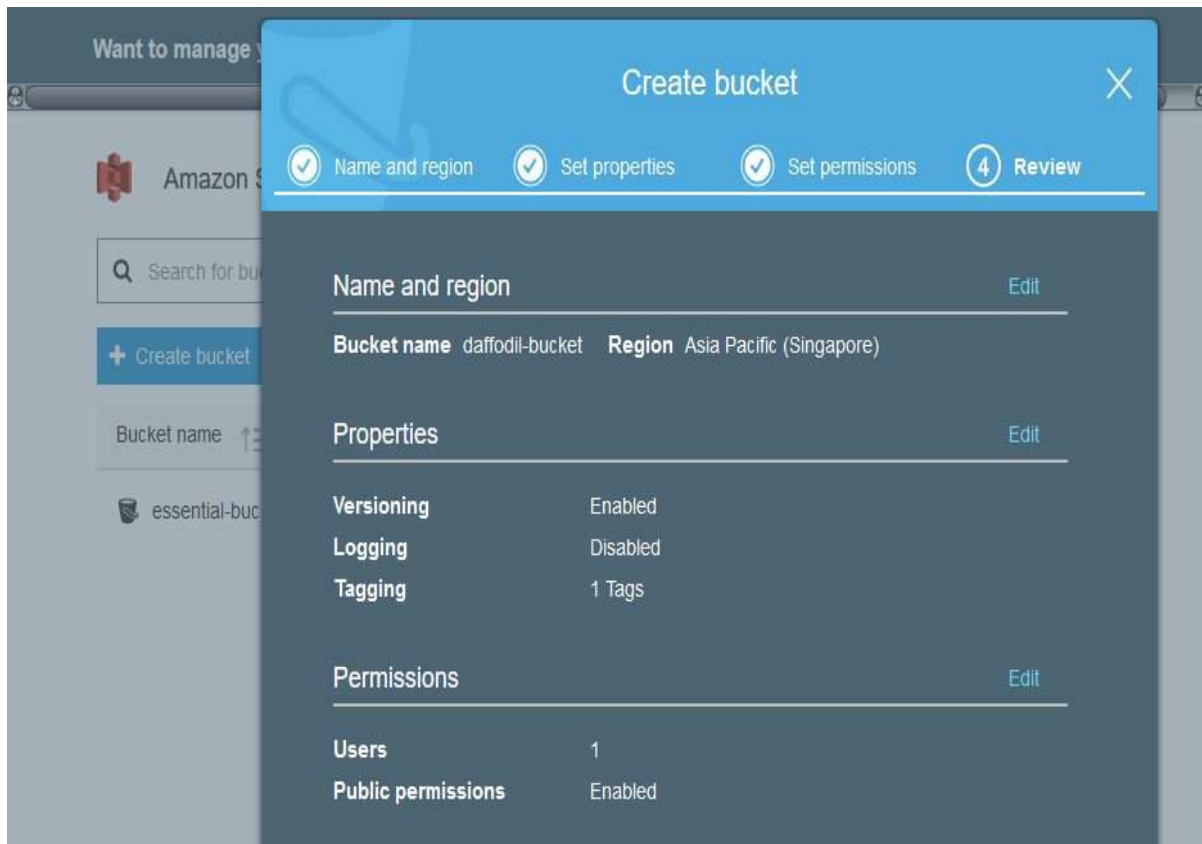


Figure 3.3.3: S3 bucket popup window

**Step5:** Now we can upload what we want into our S3 bucket by uploading files/folders.

### Pricing

Storage Cost:

Applies to data at rest in S3

Charged per Gb used

Price per GB varies based on region and storage class

Request Pricing:

- PUT
- COPY



- POST
- GET
- Data archive
- Data Restore

### 3.3.4 Relational Database Service (RDS)

RDS stands for Relational Database Service. RDS is a SQL database service that provides a wide range of SQL database options to select from.

SQL options include:

- Amazon Aurora
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQLServer

There are two main types of databases in the world:

- Relational Databases known as SQL
- Non-Relational Databases known as NoSQL

#### Provisioning RDS:

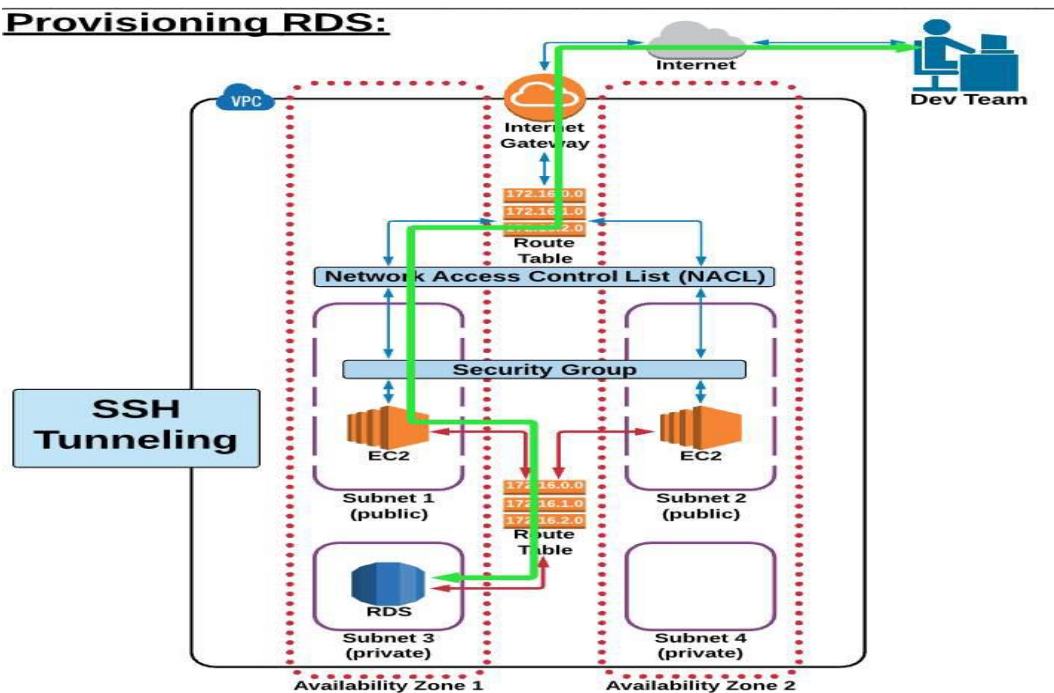


Figure 3.3.4: How RDS works

In this section, we will see how to create an RDS database and access that database through our EC2. Figure 3.3.4 showing everything. At first, we will create a SQL RDS database then connect.

#### Create RDS

**Step 1:** At first, we need to go to <https://console.AWS.amazon.com/rds/>. Then we have to create a subnet group “Daffodilsubgrooup” for our database. In this section, we will set DB Subnet Group name, Description, VPC ID, Available Zone, Subnet ID. Then create, figure 3.3.4 showing processes .

**RDS Dashboard**

- Instances
- Reserved Purchases
- Snapshots
- Parameter Groups
- External Licenses
- Option Groups
- Subnet Groups**
- Events
- Event Subscriptions
- Notifications

### Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

**Name**

**Description**

**VPC ID**

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or [add all the subnets](#) related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

**Availability Zone**

**Subnet ID**

Availability Zone	Subnet ID	CIDR Block	Action
None added			

**Figure 3.3.4: Create DB Subnet Group**

**Step 2:** Now we have to go back to our RDS Dashboard instance. Then Launched DB instance. A new section will appear in front of us. Then we can choose either Amazon Aurora or MySQL or MariaDB or PostgreSQL or Oracle database. After that, we have to select Dev/Test MySQL. Now we have to set some configuration such as DB name “MySQL”, version, DB Instance Class “t2.micro”, storage type, storage size, DB instance Identifier, Master Username, Master Password etc , processes at figure 3.3.4.

experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

☐ Only show options that are eligible for RDS Free Tier

### Create DB Instance

**DB Engine**

**License Model**

**DB Engine Version**

**DB Instance Class**

**Multi-AZ Deployment**

**Storage Type**

**Allocated Storage\***  GB

**Settings**

**DB Instance Identifier\***

**Master Username\***

**Master Password\***

**Figure 3.3.4: Create SQLdatabase 1**



**Step 3:** In this section, we have to set VPC, subnet, Availability zone, VPC Security Group for our RDS instance. This database is not publicly accessible. After that, we have to set Database Name “daffodilddb”, Database port number “3306”, DB parameter Group. Others settings will be the default. Then click Launch Db instance, shown at figure 3.3.4.

Step 1: [Select Engine](#)  
Step 2: [Production?](#)  
Step 3: [Specify DB Details](#)  
**Step 4: Configure Advanced Settings**

**Configure Advanced Settings**

**Network & Security**

VPC\* Default VPC (vpc-fdb13899)  
Subnet Group daffodilsubgroup  
Publicly Accessible No  
Availability Zone ap-southeast-1b  
VPC Security Group(s) SC (VPC), default (VPC), launch-wizard-1 (VPC), **rds-launch-wizard (VPC)**

**Database Options**

Database Name daffodilddb  
Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.  
Database Port 3306  
DB Parameter Group default:mysql5.6  
Option Group default:mysql-5-6  
Copy Tags To Snapshots ☐  
Enable Encryption No

**Backup**

Please note that automated backups are currently supported for InnoDB storage engine only if

Figure 3.3.4: Create SQLdatabase 2

**Step 4:** Now it's time to connect our database using MySQL workbench through EC2 server. Download, install and open MySQL workbench. After that click (+) sign set a connection name. At connection method section select Standard TCP/IP over SSH. After that past EC2 public IP into SSH hostname section, ubuntu as SSH Username, show EC2 private key into SSH key file section. Now it's time to set Database section. IN the field of MySQL hostname just paste the MySQL end point address “daffodilddb.cyghnya3jtez.ap-southeast-1.rds.amazonaws.com:3306” shown at figure 3.3.4.

Now it's time to test our connection. If everything ok then the test will be successful otherwise not. At last click ok,

MySQL Workbench

File Edit View Database Tools Scripting Help

**Manage Server Connections**

MySQL Connections  
DaffodilDB

Connection Name: DaffodilDB

Connection Remote Management System Profile

Connection Method: Standard TCP/IP over SSH Method to use to connect to the RDBMS

Parameters SSL Advanced

SSH Hostname: 54.169.148.197 SSH server hostname, with optional port number.  
SSH Username: ubuntu Name of the SSH user to connect with.  
SSH Password: Store in Vault ... Clear SSH user password to connect to the SSH tunnel.  
SSH Key File: C:\Users\khanik\Downloads\DaffodilKey Path to SSH private key file.  
MySQL Hostname: daffodilddb.cyghnya3jtez.ap-southeast-1.rds.amazonaws.com MySQL server host relative to the SSH server.  
MySQL Server Port: 3306 TCP/IP port of the MySQL server.  
Username: DaffodilDB Name of the user to connect with.  
Password: Store in Vault ... Clear The MySQL user's password. Will be requested later if not set.  
Default Schema: The schema to use as default schema. Leave blank to select it later.

New Delete Duplicate Move Up Move Down Test Connection Close

Figure 3.3.4: Connect MySQL Through Workbench

After everything is ok, we can create our own MySQL database. Figure 3.3.4 showing the dashboard of MySQL workbench, where we can do that.

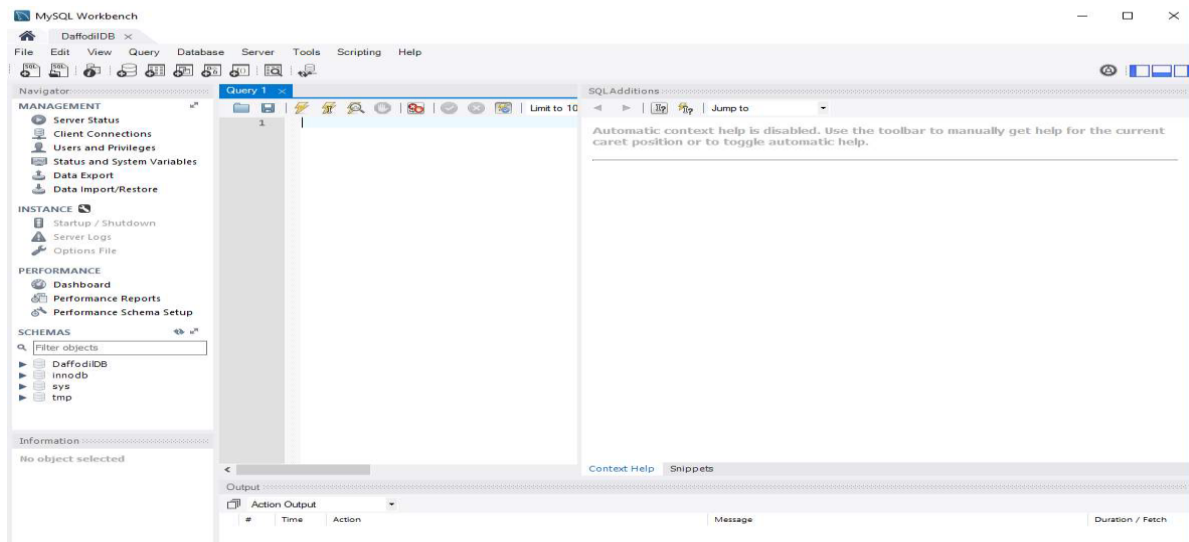


Figure 3.3.4: Connect MySQL Through Workbench 2

### Pricing

- On-Demand Instance
- Reserved Instance
- Database Storage and IOs
- Backup Storage
- Data Transfer

**3.3.5 Virtual Private Cloud (VPC)** VPC stands for Virtual Private Cloud. Where we can create our own virtual network. We can create more than one VPC at a time. In this VPC we can set up web application or database. Amazon AWS has twelve regions and every region has more than three availability zone. VPC is a private sub-section of AWS that we control, in which we can place AWS resources, for example, EC2 instance or database, in figure 3.3.5 we see our VPC with resources. We have full control over who has access to the AWS resources that we place inside our VPC [7].

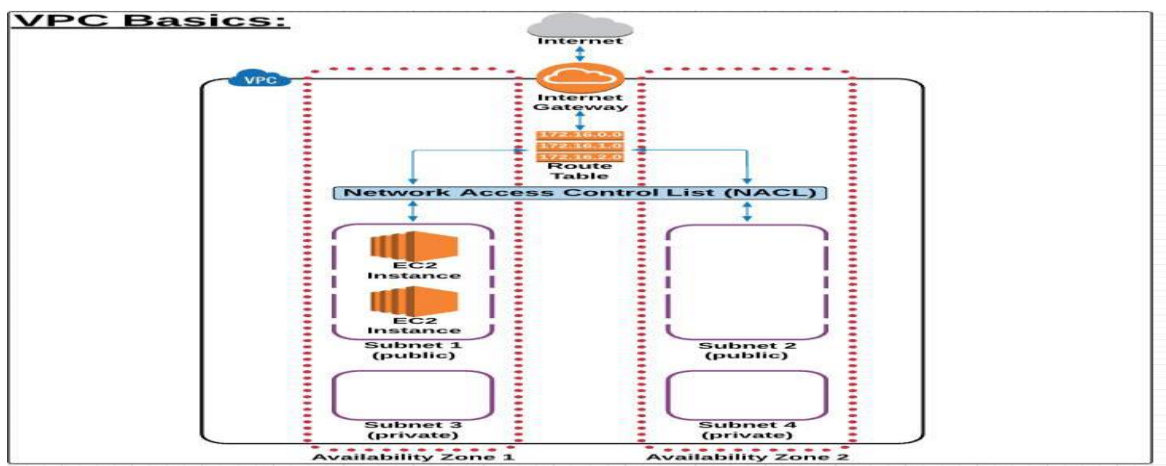


Figure 3.3.5: Virtual Private Cloud

## In VPC Section We Have

- Internet Gateway
- Route Table
- Network Access Control List (NACL)
- Subnet
- Availability Zone

### Internet Gateway

IGW or Internet Gateway is a combination of hardware and software provides our private network with a route to the world (meaning the Internet) of the VPC.

To create an IGW steps are:

**Step 1:** Go to VPC section select Internet Gateway.

**Step 2:** Create Internet Gateway

**Step 3:** A pop-up window will appear. Add tag name "DaffodilIGW".

Note:

1. Only 1 IGW can be attached to a VPC at a time.

2. An IGW cannot be detached from a VPC while there are active AWS resources in the VPC.

### Route Table

The route table is set of rules named routes. Where admin determine where network traffic is to go. Every VPC needs a route table. Without a route table network traffic won't work properly.

### To create a Route table:

**Step 1:** Go to VPC section select Route Tables.

**Step 2:** Create Route Table

**Step 3:** A pop-up window will appear. Set a tag name "DaffodilRT"

**Step 4:** Select a VPC, in which our route table will work. At last yes create

Note:

1. Unlike an IGW, we can have multiple active route tables in a VPC.

2. We cannot delete a route table if it has dependencies.

### Network Access Control List (NACL)

In a simple sentence, NACL is a security layer for our VPC, which works like a firewall for control data/packets in or out through our VPC. We can set inbound and outbound rules in NACL. Rules applied based on rule # from lowest to highest [7].

### To create an NACL:

**Step 1:** Go to VPC section select Network ACLs.

**Step 2:** Create Network ACL.

**Step 3:** A pop-up window will appear. Set a tag name "DaffodilNACL".

**Step 4:** Select a VPC, in which our NACL will work. At last yes create.

Note:

1. Rules are evaluated from lowest to highest based on rule #.

2. Any new NACL we create DENY all traffic by default.

3. A subnet can only be associated with one NACL at a time.

### Subnet

A subnet, shorthand for subnetwork, is a sub-section of a network. Generally, a subnet includes all the computers in a specific location. Circling back to the home network analogy we used in the VPC Basic lesson- if we think about our ISP begin a network, then our home network can be considered a Subnet of our ISP's network. [7]

### To create a Subnet:

**Step 1:** Go to VPC section select Subnets.

**Step 2:** Create Subnet.

**Step 3:** A pop-up window will appear. Set a tag name "Public Subnet 1/Private Subnet 1".

**Step 4:** Select a VPC, in which our Subnetwork.

**Step 5:** Select Availability Zone.

**Step 6:** set an IPv4 CIDR block "172.16.1.0/20". At last yes create.

Note:

1.Subnets must be associated with a route table.

2.A public subnet has a route to the internet.

At Last, We need to connect all thing using configuration tab. Every section has own configuration tab. Just select and set as we needed.

### 3.3.6 Identity and Access Management (IAM)

Identity and Access Management is the most important part of our security. We can deploy our own security policies in here. We can create and control our users.

We have several things to do in IAM section such as:

- Multifactor Authentication or 2MFA
- Create Users & Policies
- Setup Group & Policies
- IAM Roles

#### Multifactor Authentication or 2MFA

MFA is an abbreviation for Multifactor Authentication. It's an extra layer of security. That's how we can protect our root account get hacked. This service provided by a 3<sup>rd</sup> party company which can be free of paid service. It generates random six-digit code every couple of second when we want to log into our root account. It works via smartphone or tablet or used the app: Google authenticator, process shown at figure 3.3.6,

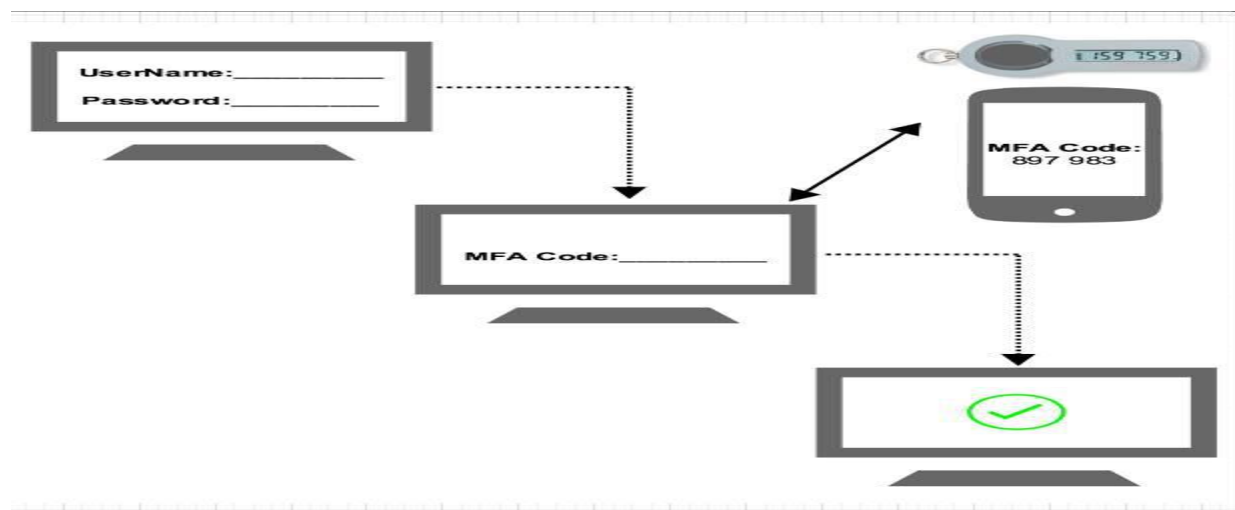


Figure 3.3.6: How Multifactor Authentication or 2MFA works

#### Setup a Multifactor Authentication

**Step 1:** Go to IAM section and select Activate MFA on your root account.

**Step 2:** Select Manage MFA. After that select the type of MFA device to activate. I choose A virtual MFA device. Select next.

**Step 3:** New pop-up window will appear. It says If you want to active this feature you have to install an application on your smartphone or PC or another device. Select next step

**Step 4:** Now will see a QR code. We need to scan this QR code by our Smartphone authentication. This time I am using Google authenticator. After that, we will give that six-digit code into authentication code box. Select Active Virtual MFA. A successful message will appear. All done. Select finish, process shown at figure 3.3.6.

Manage MFA Device

If your virtual MFA application supports scanning QR codes, scan the following image with your smartphone's camera.



[Show secret key for manual configuration](#)  
 After the application is configured, enter two consecutive authentication codes in the boxes below and click Activate Virtual MFA.

Authentication Code 1

Authentication Code 2

Cancel

Previous

Activate Virtual MFA

Figure 3.3.6: Multifactor Authentication or 2MFA QR Code

## Create Users & Policies

In this section, We will create some user and set permission or policies for those users. Let's begin:

**Step 1:** Go to IAM select Users. Select add users top left corner.

**Step 2:** Set a username. check AWS Management Console Access, check the custom password. Give a password for our user. Select next

**Step 3:** In this section, you can add this user in a group. And we are going to attach some policies by selecting attach existing policies directly. Now policies are shown, I select some policies for example AmazonEC2FullAccess, Amazons3FullAccess. You can add more or less as your wish. Select next.

**Step 4:** Here we will see all we have done some time ago just review that everything is ok. And select Create user.

**Step 5:** Now you have to download a CSV file for that user. Where you can find a link to login to AWS. Close the window. Figure 3.3.6 showing how policies works.

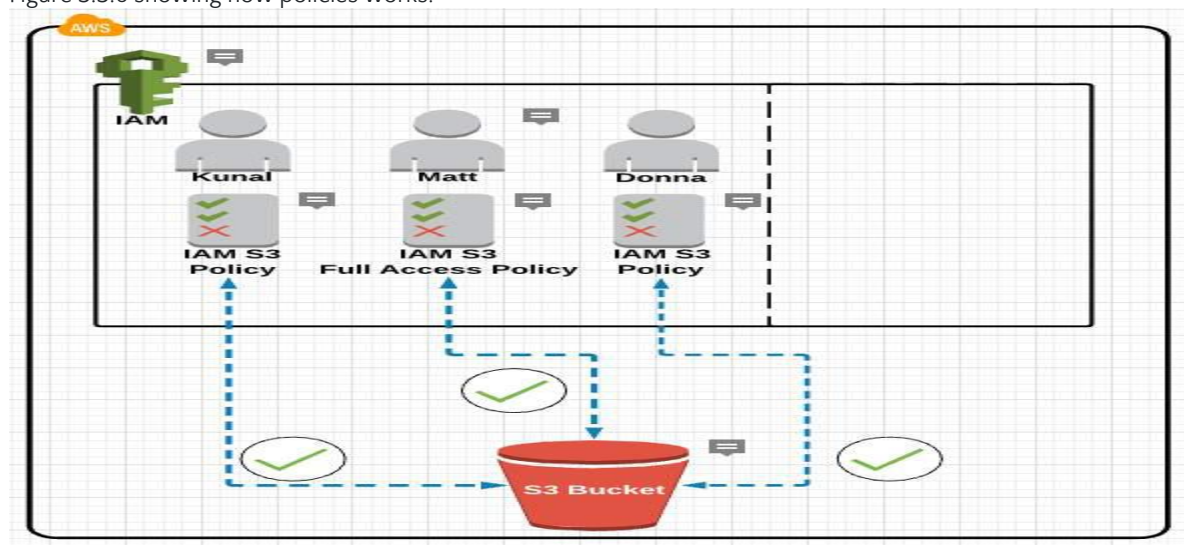


Figure 3.3.6: How user policies works

## Setup Group & Policies

In this section, we will add users into a group and set some policies for that group.

**Step 1:** Go to IAM select Groups. Select create new group top left corner.

**Step 2:** Set a group name "DaffodilGroup". select next.

**Step 3:** Attach some policies for example AmazonEC2FullAccess, Amazons3FullAccess. You can add more or less as your wish. Select next.

Figure 3.3.6 showing the working method of group policies.

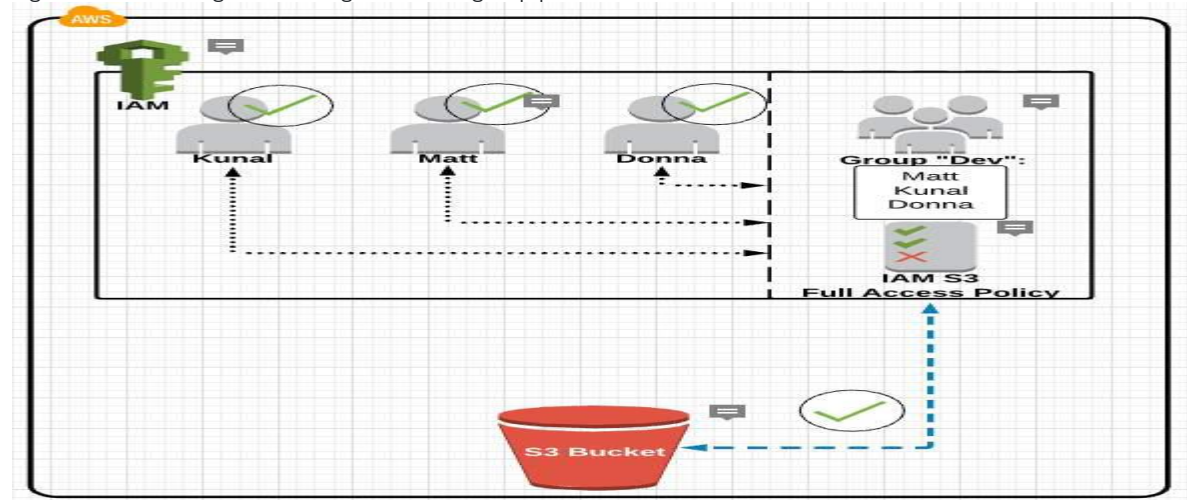


Figure 3.3.6: How Group policies works

Step 4: review and create a group.

### IAM Roles

In this section, we are going to attach two AWS service. Where one AWS service can access another service when needed.

**Step 1:** Go to IAM select Role. Select create new role top left corner.

**Step 2:** Set a group name "EC2". select next.

**Step 3:** select an AWS service which is going to attach another service.

**Step 4:** Now select your desired service I choose AmazonS3FullAccess. Select next

**Step 5:** just review and create a role. New role created.

How role works shown at figure 3.3.6 below.

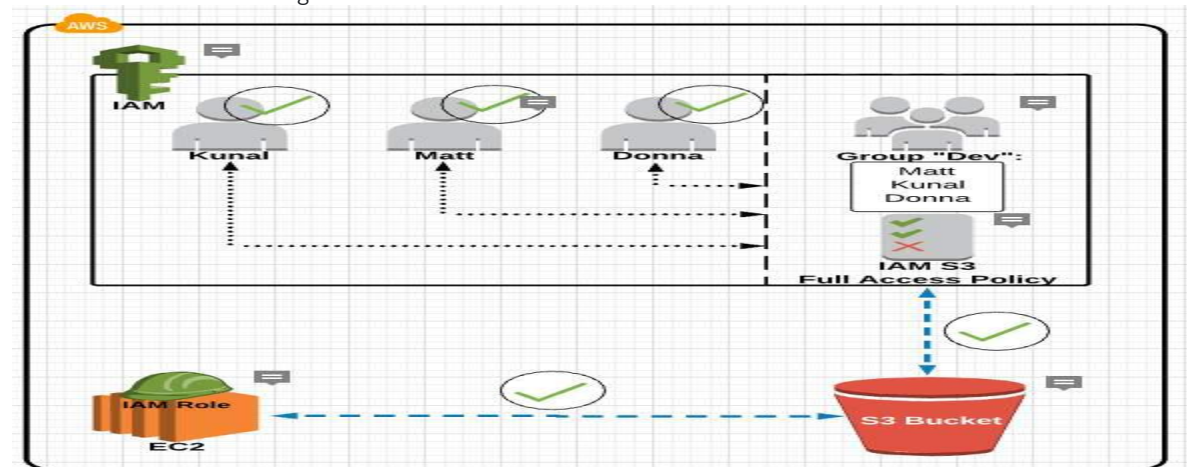


Figure 3.3.6: How Role works

### 3.3.7 Simple Notification Service (Amazon SNS)



SNS is an AWS service that allows us to automate the sending of email or text message notifications, based on events that happen in your AWS account. Using SNS service we can track our web application and send push notification through SNS to our email address, process shown at figure 3.3.7 [8].

### Create an SNS

**Step 1:** At first we have to go to Simple Notification Service. Select create topic. Now we are going to create a topic named "Auto scaling".

**Step 2:** Now give a topic name "Auto scaling". And a display name "Auto scale". Create topic.

**Step 3:** Now we need to create a subscription. New pop up appear. Change protocol to Email and give a valid email address into endpoint section. Then create a subscription.

**Step 4:** It's time to verify the submitted email. AWS sent a mail to our mail address. Just click the link Confirm Subscription. Go back to the SNS and see a subscriber number.

**Step 5:** Now we have to publish our topic. So click publish topic, add a subject and add some text in the message field. This message will be sent to the subscriber. So be careful about it. Select Publish Message.

### SNS Usage Example/Workflow:

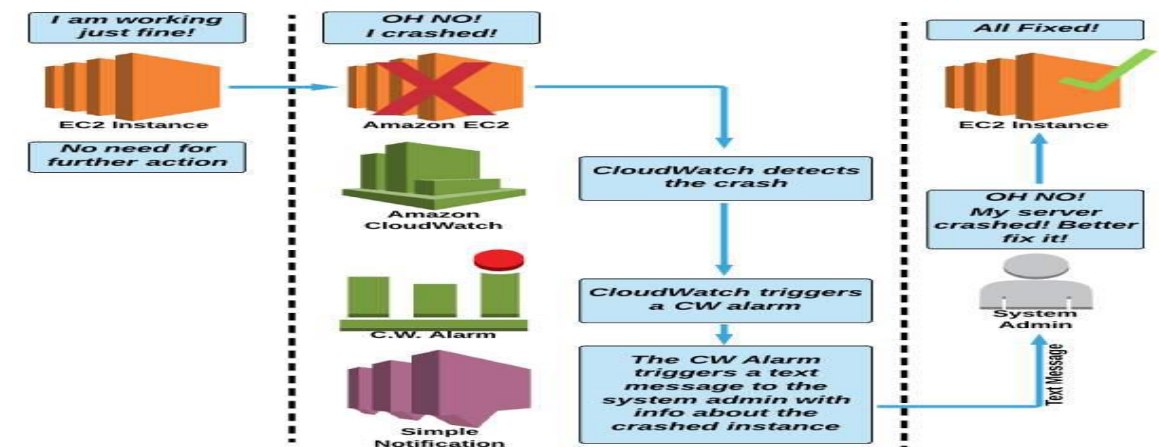


Figure 3.3.7: How SNS works

### 3.3.8 Cloud Watch

CloudWatch is a service that allows us to monitor various elements of our AWS account. CloudWatch monitors our real-time resources deployed into Amazon AWS. Using CloudWatch matrices we can measure our cloud applications. CloudWatch set alarms and send notifications of the resource that we are monitoring, process shown at figure 3.3.8 below [9].

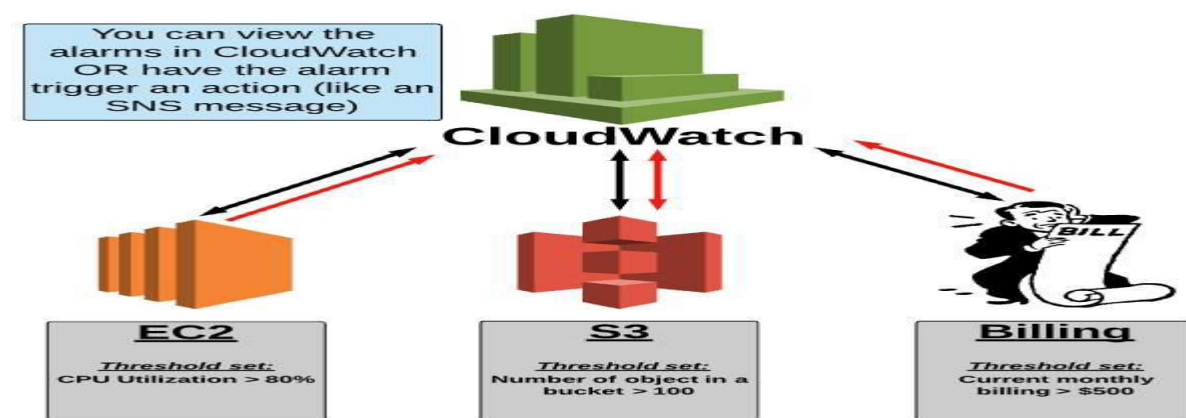


Figure 3.3.8: How CloudWatch works

## Create a Dashboard

**Step 1:** Go to CloudWatch and select Dashboard. Then Create a Dashboard, give it a name "DaffodilDashboard". Then create a dashboard.

**Step 2:** After that choose a widget for the dashboard.

**Step 3:** Explore the available metrics and select metrics that you want.

**Step 4:** Now create the widget.

## Create an Alarm

**Step 1:** Select a Dashboard and go to alarm section,

**Step 2:** create alarm, then select a category and next, figure 3.3.8 showing CPU utilization dashboard.

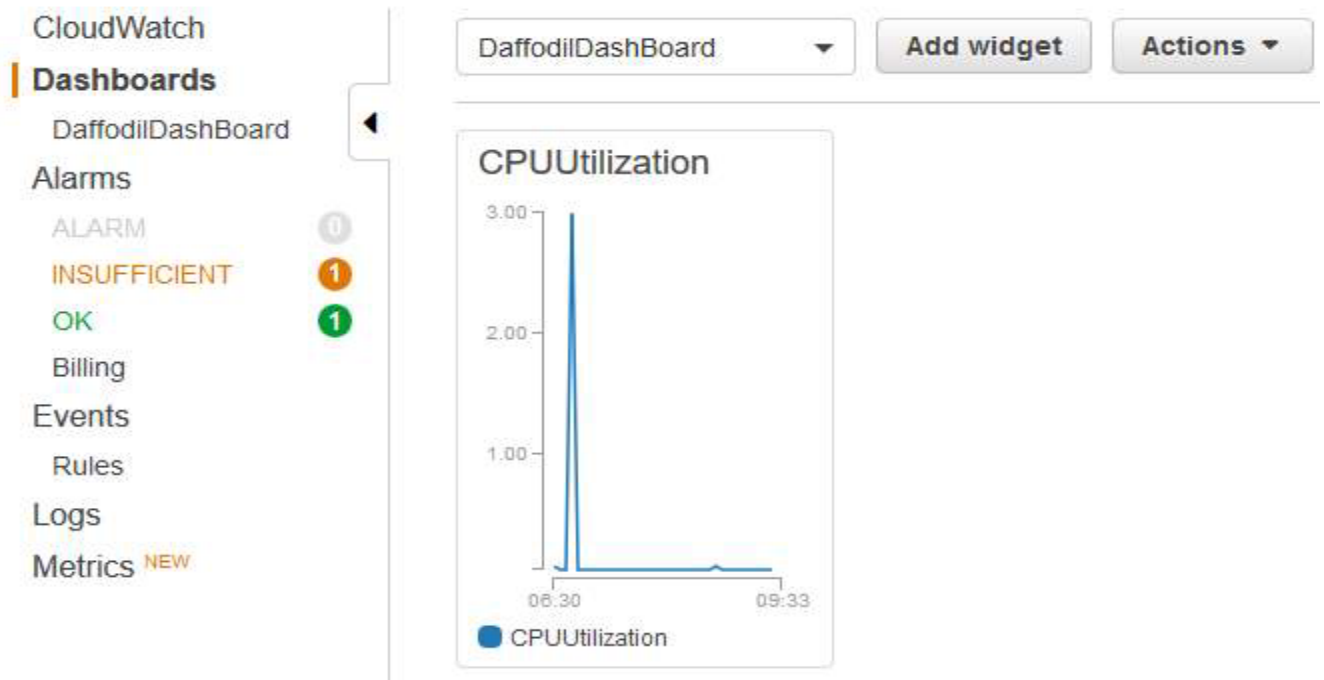


Figure 3.3.8: CloudWatch status

**Step 3:** Now give a name and description of this alarm. And set some metrics. Such as CPU utilization is  $\geq 30\%$  for 5 consecutive periods. Select whenever this alarm and send a notification to. At last, create alarm.

## Pricing

- Amazon CloudWatch Dashboard
- Detailed Monitoring for EC Instance
- Amazon CloudWatch Custom Metrics
- Amazon CloudWatch Alarm
- Amazon CloudWatch Logs
- Amazon CloudWatch Custom Events

## 3.3.9 Elastic Load Balancing

Elastic Load Balancer evenly distributes web traffic between EC2 instances that are associated with it. ELB equally distribute incoming web traffic to multiple EC2 instances which are located in multiple Availability Zones. Fault tolerance is one of the most vital features of Elastic Load Balancer. When one Ec2 will crash or down then ELB pass web traffic to another EC2 instance, process shown at figure 3.3.9. That's how our WEB server or application won't be offline never [10].



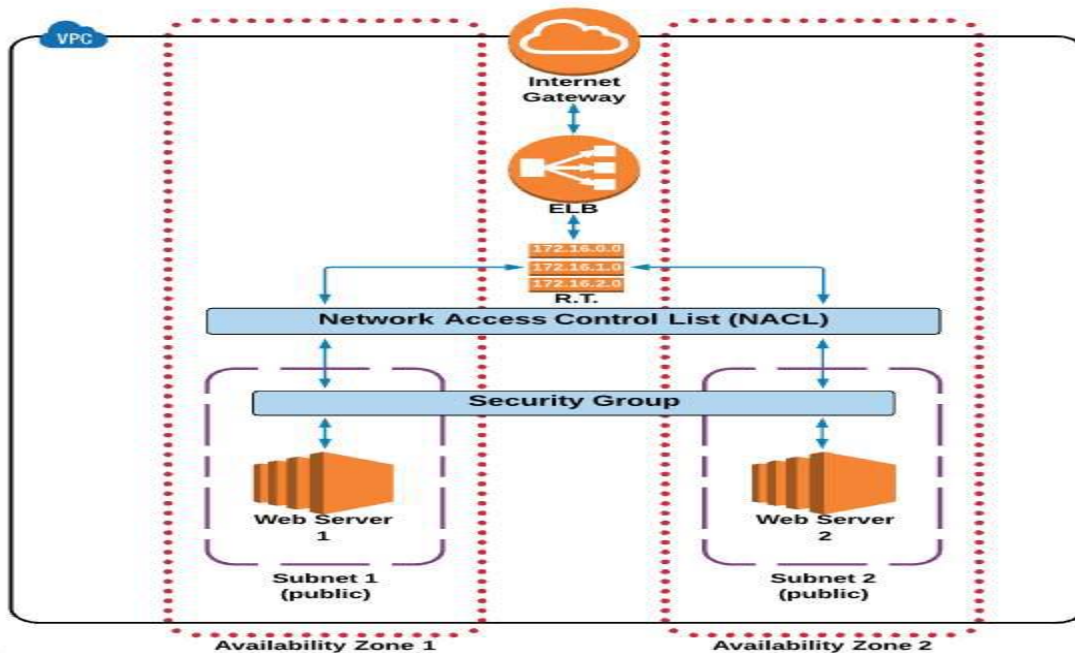


Figure 3.3.9: How ELB works

### Create Elastic Load Balancer

**Step 1:** At first, we have to go to EC2> Load Balancer

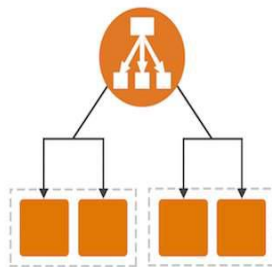
**Step 2:** Create Load Balancer.

**Step 3:** A new popup window will appear. Just select Classic Load Balancer and continue, shown at figure 3.3.9 below.

Welcome to Elastic Load Balancing

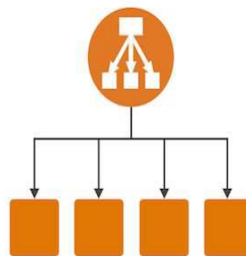
☐ Application Load Balancer

☒ Preferred for HTTP/HTTPS



An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each EC2 instance or container instance in your VPC.

☒ Classic Load Balancer



A Classic Load Balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS), and supports either EC2-Classic or a VPC.

Cancel Continue

Figure 3.3.9: Classic Load Balancer

**Step 4:** Give a name, select a VPC for our ELB. Add some protocols. Next.

**Step 5:** Create or add existing security Group for ELB. Next.

**Step 6:** Now it's time to configure health check. we are using TCP protocol and 80 port. Next, process shown at figure 3.3.9 below.

1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review

### Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol: TCP  
 Ping Port: TCP

Advanced Details

Response Timeout: 5 seconds  
 Interval: 30 seconds  
 Unhealthy threshold: 2  
 Healthy threshold: 10

[Cancel](#) [Previous](#) [Next: Add EC2 Instances](#)

Figure 3.3.9: ELB health check

**Step 7:** Here we are going to add EC2 instance for ELB. Next

**Step 8:** here you can add a tag or not. Next

**Step 9:** Now it's review time. If everything ok then clicks Create. Figure 3.3.9 showing how our ELB distribute our web traffic to multiple web server.



Figure 3.3.9: ELB working method

### 3.3.10 Auto Scaling

Auto Scaling automates the process of adding or removing EC2 instances based on traffic demand for our application. Auto scaling is one of the best-awarded Innovation of Amazon AWS. Using this service, we can deploy a minimum number of the instance at a time because of our system never goes down. Also, we can deploy a maximum number of the instance when we need those instances will be active shown at figure 3.3.10 below [11].



Figure 3.3.10: How auto-scaling works

### Create Auto Scaling

**Step 1:** At first, we have to go to EC2> Launch Configuration.

**Step 2:** Select an AMI "Ubuntu 16.04".

**Step 3:** Select an Instance Type "t2. micro".

**Step 4:** Create Launch Configuration. set a name "auto scaling". Set a public IP for our instance, process shown at figure 3.3.10 below.

Figure 3.3.10: Launch Configuration of auto scaling

**Step 5:** Select storage type and add storage if needed.

**Step 6:** Configure Security Group. I make new one and named "DaffodilSG", VPC, Subnet, process shown at figure 3.3.10 below.

**Create Auto Scaling Group**

**Launch Configuration**

Group name: DaffodilASG

Group size: Start with 2 instances

Network: vpc-fdb13899 (172.31.0.0/16) (default)

Subnet: subnet-f6661880(172.31.16.0/20) | Public Subnet 2 | Default in ap-southeast-1b

**Advanced Details**

**Load Balancing**

☒ Receive traffic from one or more load balancers

**Classic Load Balancers**

DaffodilELB

**Target Groups**

**Health Check Type**

☒ ELB ☐ EC2

**Health Check Grace Period**

300 seconds

**Monitoring**

Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are

**Next: Configure scaling policies**

Figure 3.3.10: Create Auto Scaling Group

**Step 7:** Review your configuration and create.

**Step 8:** Now we have to create Auto Scaling Group. Give a Group name “DaffodilASG”. Set group size. I set 2. After that, you have to select a VPC for auto scaling group. Also, need to add subnets. IN advanced option we have to set “DaffodilELB” into Classis Load Balancer field. And check ELB in health check section. Next

**Step 9:** Here we have to set minimum and maximum instance we want for auto scaling. After that click adds a new alarm in Increase Group Size section and set some metrics. Do the same thing for Decrease Group Size. I set some metric that’s why when EC2 CPU utilization goes over 70% it will create one more instance automatically. And when CPU utilization goes below 30% one instance will be terminated automatically, process shown at figure 3.3.10 below.

**Create Auto Scaling Group**

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

☐ Keep this group at its initial size

☒ Use scaling policies to adjust the capacity of this group

Scale between 2 and 3 instances. These will be the minimum and maximum size of your group.

**Increase Group Size**

Name: Increase Group Size

Execute policy when: awsec2-DaffodilASG-High-CPU-Utilization

breaches the alarm threshold: CPUUtilization >= 30 for 300 seconds for the metric dimensions AutoScalingGroupName = DaffodilASG

Take the action: Add 0 instances when CPUUtilization <= 30

**Review**

Figure 3.3.10: Increase of auto scaling group

**Step 10:** After that, we will configure Decrease auto scaling section where I set when CPU utilization down below 30% then our instance will be terminated, process shown at figure 3.3.10 below.

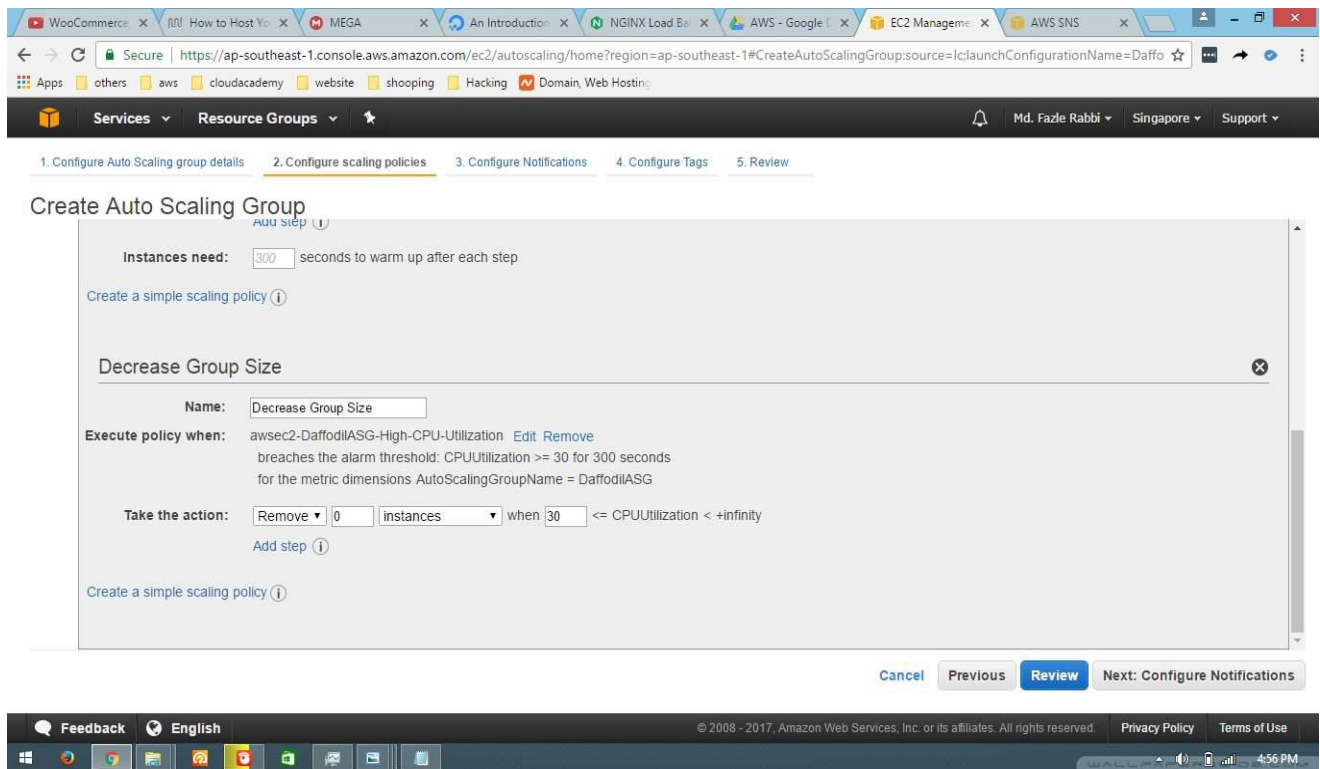


Figure 3.3.10: Decrease auto scaling group

**Step 10:** Here we will add an SNS topic to send a notification to admin. Then admin will check the instance and will take the necessary steps, process shown at figure 3.3.10 below.

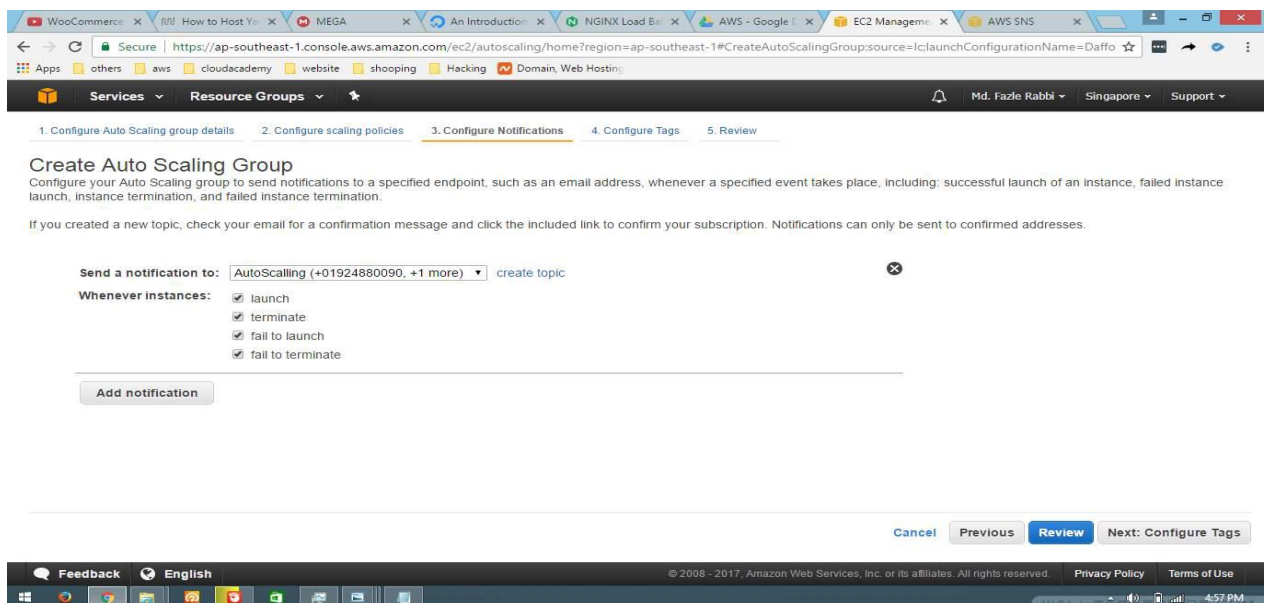


Figure 3.3.10: Configure notification of auto scaling



Step 11: Add a tag name.

Step 12: Review and Launch our instance. Now we will see our new instance is begun start to run into EC2 instance section.

Figure 3.3.10 is showing how our auto-scaling works in real life.



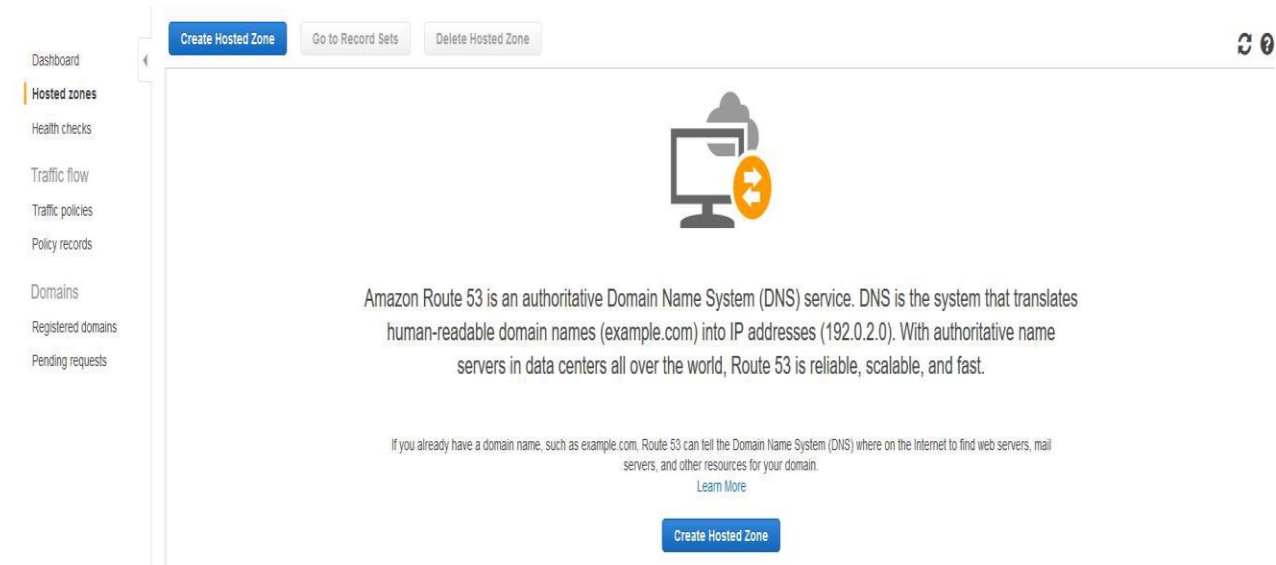
Figure 3.3.10: Auto-scaling working methods

### 3.3.11 Amazon Route 53

Route 53 is where we configure and manage web domains for websites or applications we host on AWS. In Route 53 we can Register a new domain, use DNS service and also can health check. In this section, we can do traffic management and availability monitoring.

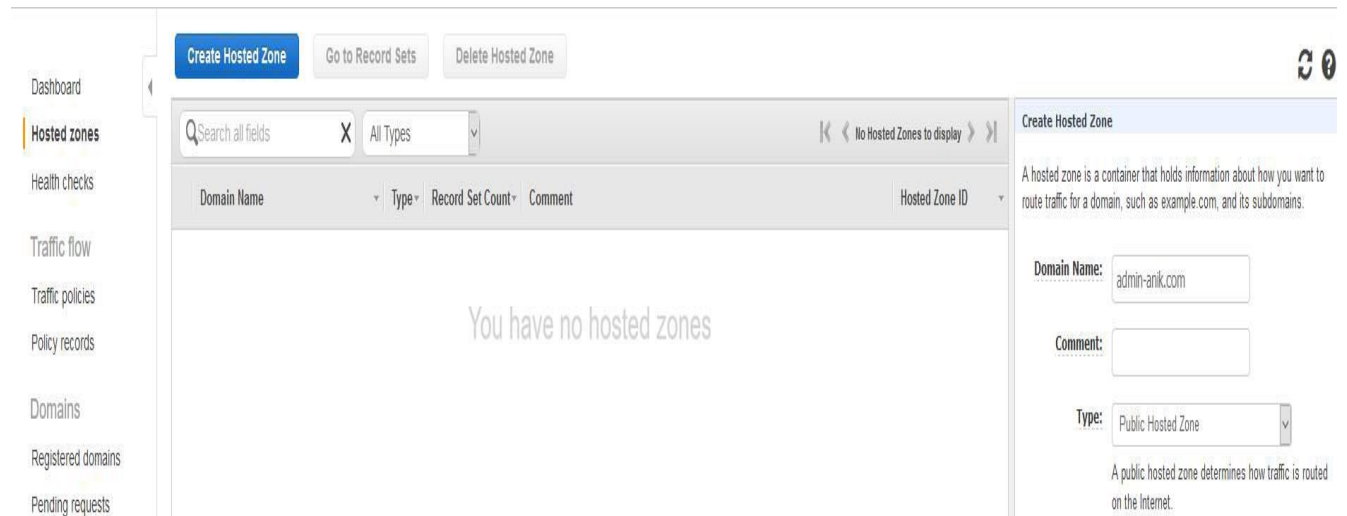
#### Create Route53

Step 1: At first, we have to go to Route53> Hosted zones , process shown at figure 3.3.11



**Figure 3.3.11: Route 53 Hosted zones**

**Step 2:** Create Hosted Zone, a new popup will open. Add a domain name “admin-anik.com” and select as Public Hosted Zone from the drop down menu, check right corner of the figure 3.3.11 below.



**Figure 3.3.11: Add domain name**

**Step 3:** A hosted zone created. Now we are seeing some NS records and an SOA record. Which are very much important for every site. Now we are going to add some A records.

At first click on Create Record Set, then

Name: www

Type: A – Ipv4 address,

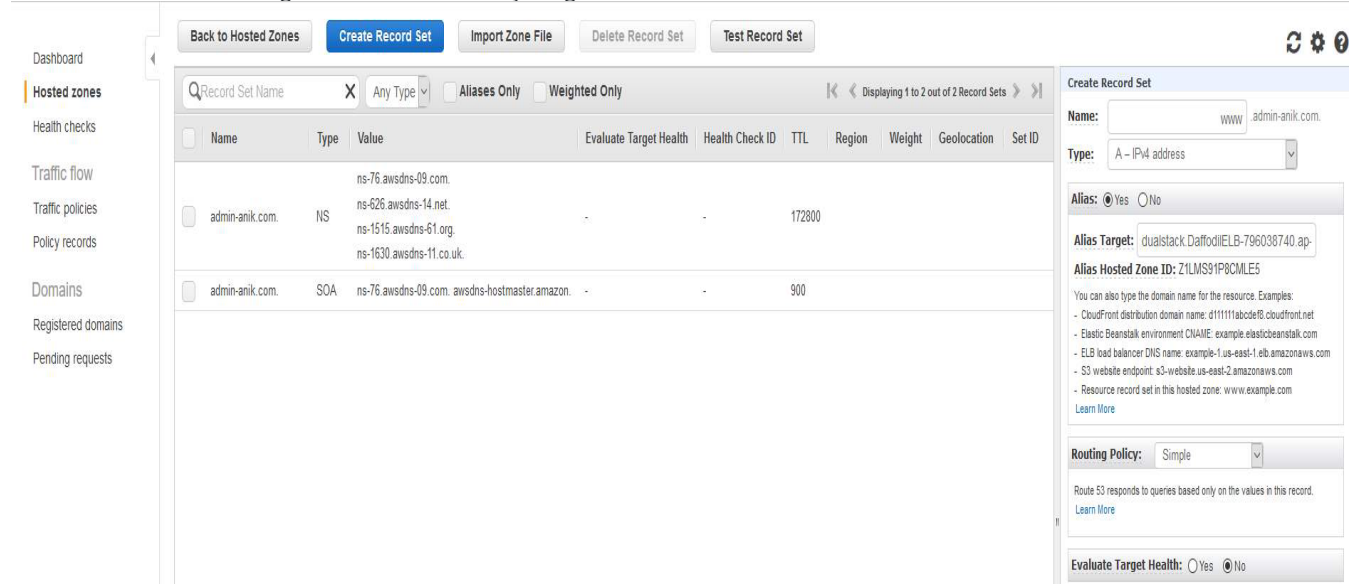
Alias: Yes,

Alias Target: select DaffodilELB

Routing policy: simple

Evaluate Target Health: no

Then click create. Check figure 3.3.11 below, everything is in it.



**Figure 3.3.11: Set A record**

**Step 4:** Now another A record needs to setup:  
 The first click on Create Record Set, then  
 Name:  
 Type: A – Ipv4 address,  
 Alias: Yes,  
 Alias Target: select DaffodilELB  
 Routing policy: simple  
 Evaluate Target Health: no  
 Then click create. Check figure 3.3.11 below.

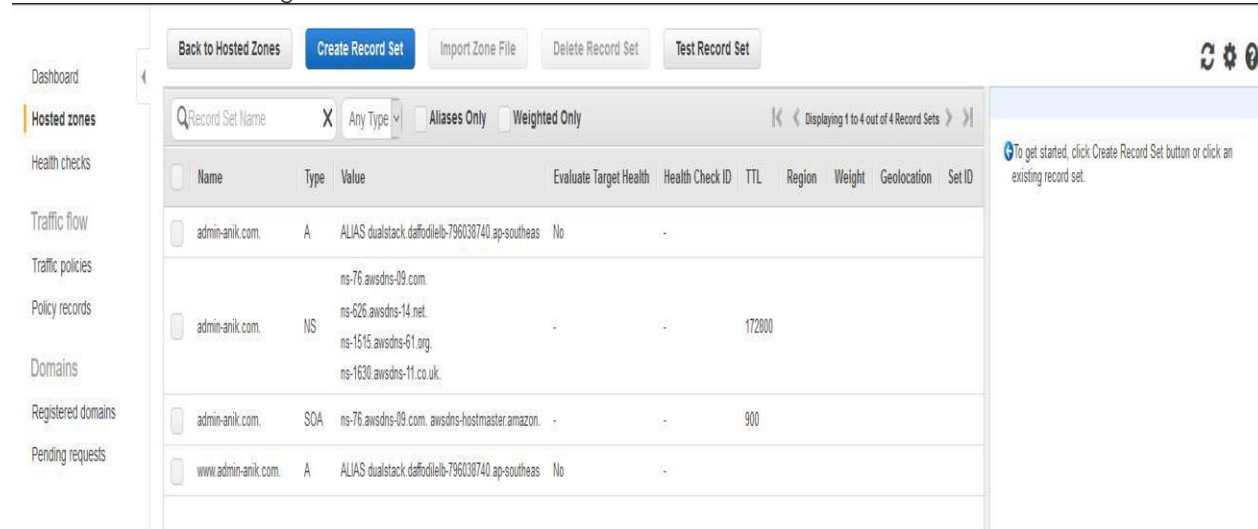


Figure 3.3.11: Route 53 DNS records

**Step 5:** Now we have to go to where we bought our Domain. I bought my Domain from Namecheap. Select Domain list then select Domain. Then go to Nameserver select Custom DNS. After that add 4 DNS record which was given by Amazon AWS. Mine was  
 ns-76.AWSDns-09.com.  
 ns-626.AWSDns-14.net.  
 ns-1515.AWSDns-61.org.  
 ns-1630.AWSDns-11.co.uk.

Figure 3.3.11 showing how to configure namecheap DNS below.

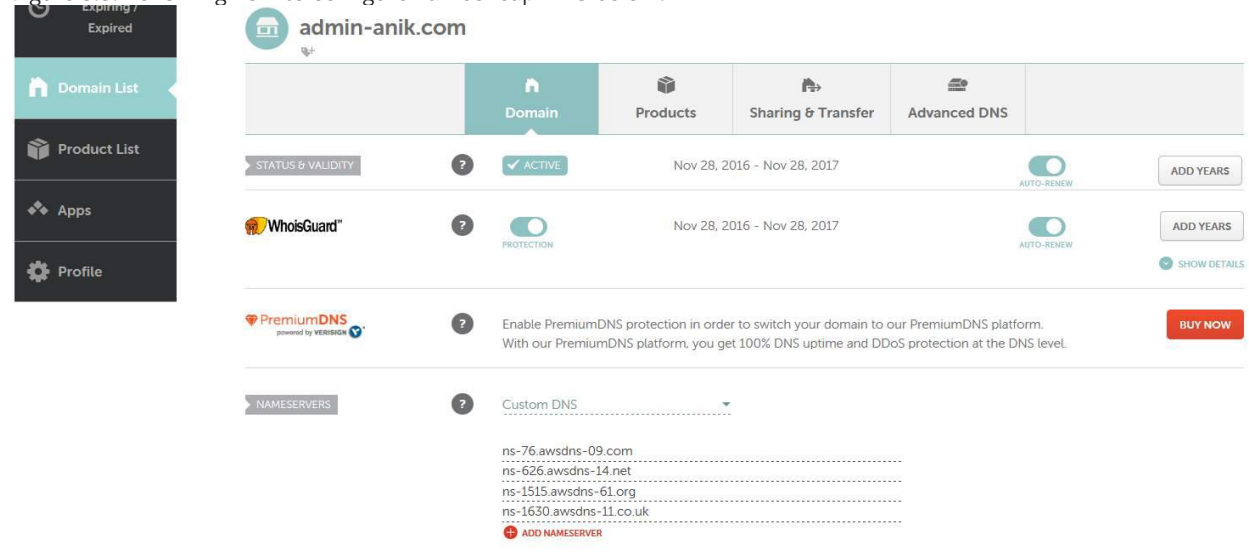


Figure 3.3.11: Name cheap DNS configure



After that click ok. It will take some time to change DNS server of our domain. SO be patient. Then type your domain name into the browser, Yes your site is live now. See figure 3.3.11 below.

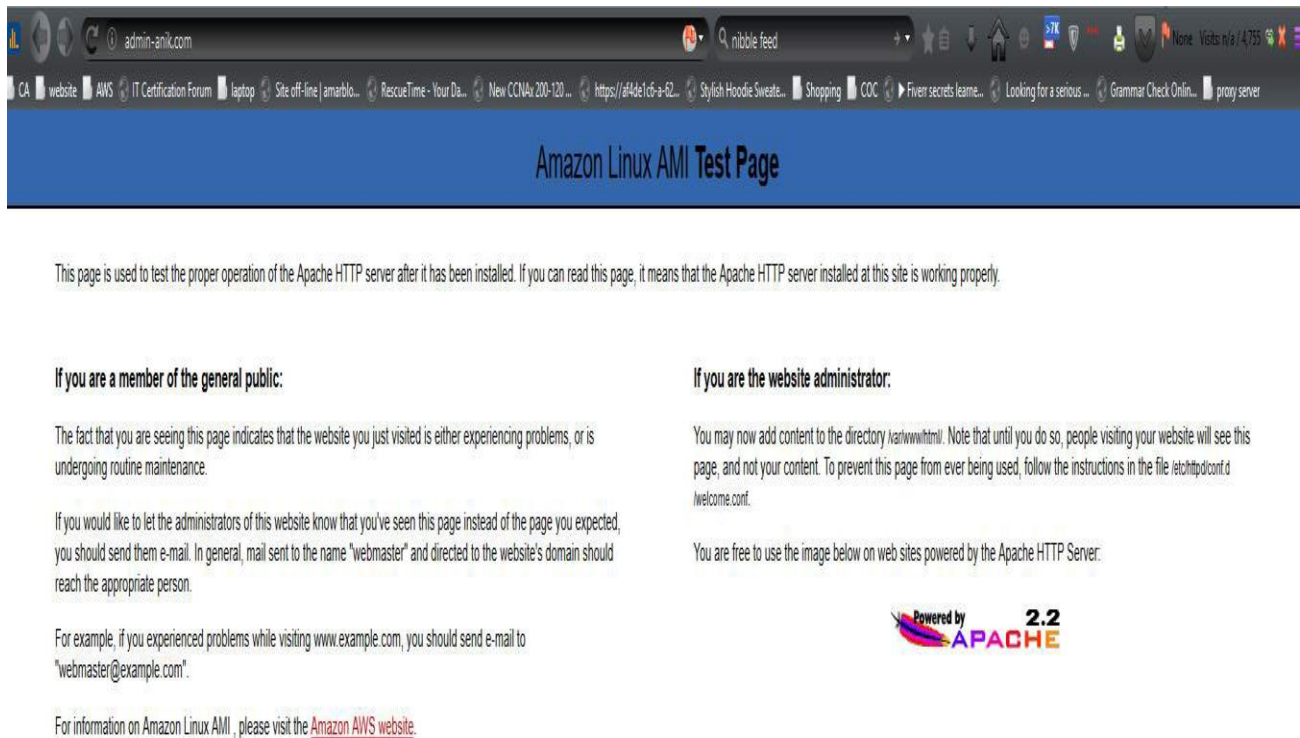


Figure 3.3.11: Site Live

### 3.4 Challenges

I faced many problems while working in cloud and making this report. I am using AWS free account, so there are many limitations to using their services. I can only use some basic services to develop a highly available, cost effective, fault tolerance scalable system. I also used the lucid chart to create UML Diagram and it is also a free account where is some limitation. I can create only 60 objects using a free account. I made my project angle of Service Providers because of my company only give those types of services. I made this system for the project only but it was not build based on any person or organization requested requirements.

## CHAPTER 4: COMPETENCIES AND SMART PLAN

### 4.1 Competencies Earned

During this internship, I achieved many new skills which are very important for my future career.

- How to build a cloud architecture
- How analysis a cloud system
- Project management skill
- Server deployment and Maintenance
- System Automation.
- Security

### 4.2 Smart Plan

The growth of cloud computing continues at a phenomenal rate – and as a result, the employer demand for cloud professionals has exploded. In 2015, there were an estimated 18 million cloud computing jobs globally, according to WANTED Analytics, and it's a number one [hottest skillset in 2016](#).

So, it will be a good plan to build career as a Cloud Computing engineer. There are huge opportunities in our country. Some company recruiting Cloud Computing engineer. So, this is the 1<sup>st</sup> generation of cloud computing in Bangladesh. This is time to build a career as a Cloud Engineer.

Within 1 year I want to complete two AWS cloud certification course. And those are:

AWS Solutions Architect Associate  
AWS Certified Developer – Associate

### 4.3 Reflections

**What tools did you use or learned to use?**

I used putty as a terminal to access my cloud servers, puttyGen to use .pem and .ppk file. I used lucid chart for designing a cloud system. And MySQL Workbench to access cloud database through SSH [14].

**What has DSP was done that has helped you obtain or better prepare yourself for your internship?**

It really helped me to develop myself in terms of communicating effectively and concisely. I work in a fast-paced environment where constant communication with my team and other departments are crucial to project success [14].

**What did you struggle with throughout your internship?**

I struggled with learning Server configuration in ubuntu and built a system based on cloud services. It was very difficult for me to adapt with cloud within a short time [14].

**What advice would you give to someone in your field?**

In my field, you must be able to adapt to the ever-changing technologies. Because Every month's new cloud services are coming. You have to be patient. Take your time and keep learning about cloud computing [14].

## CHAPTER 5: CONCLUSION AND FUTURE CAREER

### 5.1 Discussion and Conclusion

Cloud computing is a newly developing paradigm of distributed computing. Virtualization in combination with utility computing model can make a difference in the IT industry and as well as in social perspective. Though cloud computing is still in its infancy but it's clearly gaining momentum. Organizations like Google, Yahoo, and Amazon are already providing cloud services. The products like Google App-Engine, Amazon EC2, and Windows Azure are capturing the market with their ease of use, availability aspects, and utility computing model. Users don't have to be worried about the hinges of distributed programming as they are taken care of by the cloud providers [15].

Finally, to guarantee the long-term success of Cloud Computing, the chapter tackles some significant challenges that face the Cloud paradigm. Challenges that need to be carefully addressed for future research like; user privacy, data security, data lock-in, availability, disaster recovery, performance, scalability, energy efficiency, and programmability [15].

### 5.2 Scope for Further Career

Cloud Computing is the most demandable IT sector now. Cloud Engineer has a bright future. According to the statistics provided by the Market Research Media, the worldwide market for Cloud Computing is likely to grow at a CAGR of 30% to reach US\$ 270 billion by the year 2020. Considering the cutting-edge innovations and new industry-specific applications, Cloud Computing is fast emerging as an essential component of an enterprise's IT framework [16].

Cloud computing is good for both big and small organization that's why they have deployed the cloud technology in some suitable capacity. Enterprises need more IT professionals to work around 'the cloud'. The Cloud Computing industry requires professionals with adept training and knowledge in both technical and managerial fields. The demand for IT professionals continues to rise at an exponential rate as more and more enterprises adopt Cloud Computing [16].

Here are some of the popular Cloud related job profiles:

- Cloud Software Engineer
- Cloud Project Manager
- Cloud Business Analyst
- Cloud Network Architect/Planner
- Cloud Product Manager
- Cloud Sales Executive
- Cloud [Developer/Programmer](#)
- Cloud Consultant
- Cloud Systems Engineer
- Cloud Systems Administrator
- Cloud Network Engineer

The demand for professionals with knowledge of Cloud Computing is expected to rise exponentially because more and more companies are implementing this technology.

## References

- [1] Dr. Birendra Goswami Usha Martin Academy, Ranchi,&Dr. S.N.Singh XISS, Ranchi Abstracts-Seminar on Cloud Computing 22.11.12
- [2] Company overview << <https://www.previewtechs.com/about> >> last accessed on 02-04-2017 at 1:00am.
- [3] Preview thechnologies << <https://www.previewtechs.com> >> last accessed on 02-04-2017 at 2:00am.

- [4] Amazon Elastic Compute Cloud "Documentation" available at  
<< <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html> >>
- [5] Amazon Simple Storage Service "Documentation" available at  
<< <http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html> >> last accessed on 02-04-2017 at 2:30am.
- [6] Amazon Virtual Private Cloud "Documentation" available at  
<< <http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/getting-started-ipv4.html> >> last accessed on 02-04-2017 at 2:45am.
- [7] Amazon Relational Database Service "Documentation"  
<< <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html> >> last accessed on 02-04-2017 at 3:55am.
- [8] Amazon Simple Notification Service "Documentation"  
<< <http://docs.aws.amazon.com/sns/latest/dg/welcome.html> >> last accessed on 02-04-2017 at 10:00am.
- [9] Amazon CloudWatch "Documentation" available  
at << <http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html> >> last accessed on 02-04-2017 at 10:15am.
- [10] Elastic Load Balancing "Documentation" available at  
<< <http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html> >> last accessed on 02-04-2017 at 10:25am.
- [11] Auto Scaling "Documentation" available at  
<< <http://docs.aws.amazon.com/autoscaling/latest/userguide/WhatIsAutoScaling.html> >> last accessed on 02-04-2017 at 10:40am.
- [12] Using nginx as HTTP load balancer, documentation, available at  
<< [http://nginx.org/en/docs/http/load\\_balancing.html](http://nginx.org/en/docs/http/load_balancing.html) >> last accessed on 02-04-2017 at 12:00pm.
- [13] Nginx Load Balancing HTTP Load Balancer, documentation, available at  
<< <https://www.nginx.com/resources/admin-guide/load-balancer> >> last accessed on 02-04-2017 at 12:45pm.
- [14] Internship Reflection available at << <http://www.dspsjsu.org/internship-reflection> >> last accessed on 02-04-2017 at 3:00pm.
- [15] Abhirup Ghosh, Cloud Computing, Seminar Report , 11,5, 2015
- [16] Cloud Computing and its Scope in Future available a