A MINOR-PROJECT REPORT

ON

PlaceComms: Smart Online Recruitment Platform

Submitted To

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY – KIIT

(Deemed to be University, Established U/S 3 of UGC Act, 1956)

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

BY

Himanshu Kumar	**1521018**
Rishabh	**1521030**
Rashmi	**1521027**
Anjali Gupta	**1505586**

UNDER THE GUIDANCE OF

Prof. Roshini Pradhan

(Asst. Professor (I), School of Computer Engineering)

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

2017-2018

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY - KIIT

(Deemed to be University, Established U/S 3 of UGC Act, 1956)

School of Computer Engineering

Bhubaneswar, Odisha – 751024

CERTIFICATE

This is to certify that the project entitled

PlaceComms: Smart Online Recruitment Platform

submitted by

Himanshu Kumar	**1521018**
Rishabh	**1521030**
Rashmi	**1521027**
Anjali Gupta	**1505586**

is a record of bonafide work carried out by them, in partial fulfilment of the requirement for the award of Degree of Bachelor of Technology (Computer Science and Engineering) at KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY – KIIT (Deemed to be University), Bhubaneswar. This work is done during the year 2017-2018 under our guidance.

Date: / /

Prof. Roshni Pradhan

Project Guide

ABSTRACT

The aim of this project is to build a highly Fault-Tolerant, Scalable, Secure, Cross-Region VPC based Online Recruitment Platform running fully on efficiently configured AWS stack backed by technologies like Laravel and Mongodb. These are achieved by deploying 30 T2 Burstable Performance Instances over 3 Availability Zones and 10 Aws Regions for the Apache Servers and 3 T2 Burstable Performance Instance over 3 Availability Zones and 1 Aws Region for the Mongodb Database with Elastic Load Balancers in all the regions.

This is an unique and effective online recruitment platform with features like on demand server scaling, burst mode and load balancers for heavy traffics, server speed based user dns routing, automatic instance backup for quick recovery, dedicated Mongodb clusters for quick database operations, a periodic job monitors CloudWatch metrics for the on-demand nodes, Laravel which provides up capability to speed up development cycles, foster innovation and maintain ownership and improve maintainability and scalability of this software application. Our implementation of AWS Stack is fully custom designed with all the security and speed aspects in mind, which can also be used for any type of large scale online application capable of processing thousands of requests per second.

ACKNOWLEDGMENTS

We are profoundly grateful to **Prof Roshini Pradhan** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

Himanshu Kumar	**1521018**
Rishabh	**1521030**
Rashmi	**1521027**
Anjali Gupta	**1505586**

Contents

Chapter 1

INTRODUCTION

1.1. Live Implementation of Placecomms

Internship Camp' 18, which was held from 23rd to 25th of March 2018 is the flagship implementation of our customized AWS stack. It aims to provide internship opportunities to undergraduate and postgraduate students from the field of technology, biotechnology, management, designing and law from Bhubaneswar and our home university. This unique platform ensures that the companies get the most young and creative brains who can add some value to their organisation. Simultaneously the students also get to implement their knowledge in solving real life problems and gaining work experience. The internship camp also provides a networking platform to all the start-up and corporate firms to exchange ideas and increase their network.

1.2. Purpose

The functions and resources available to an online ecosystem have changed a lot since the early times of static html pages. Recently we've seen many significant advancements: simple flexible layouts (finally), streaming video, real 3D graphics and native animations, to name a few. But there is obviously still more to do, especially if the internet is to become a complete effective system with native platforms.
Developers and browser engineers have been looking at the features that make their platform and apps so popular, after seeing how the internet matches up, and working continuously to fill any holes in the web platform's capabilities. Essentially it comes down to implementing awesome set of killer features, some of which are simple to implement, but others not so much

.

1.2.1 Amazon Web Services

Amazon Web Services (AWS) is a subsidiary of Amazon Inc that provides a large collection of infrastructure and cloud application services that enable you to run almost everything in the cloud – from large enterprise applications to massive data projects and faster mobile apps.AWS is the industry’s cloud market leader, with customers spread over almost every country over the globe, and a huge part of the Cloud Infrastructure as a Service (IaaS) market.

1.2.2 Why we choose AWS ?

AWS has an amazing active customer base of over 190 countries, and more than a million active customers – including nearly 5,000 education institutions, 2,000 government agencies, and more than 17,500 non-profits still counting. The AWS global infrastructure is made up of 33 availability zones across 12 geographic regions worldwide, with 5 more addition regions were been added in 2016.
Today, more than one-third of all Internet users visit a site or application powered by AWS. AWS data centers are located in the India, U.S., Europe, Singapore, Japan, and Australia. AWS scales up capacity by adding new data centers – with the idea that customers are architecting their applications for local data protection.

1.2.3 Amazon Services Used:

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates our need to invest in hardware up front, so we can develop and deploy applications faster. We can use Amazon EC2 to launch as many or as few virtual servers as we need, configure security and networking, and manage storage. Amazon EC2 enables us to scale up or down to handle changes in requirements or spikes in popularity, reducing our need to forecast traffic.

1.2.4 Features of Amazon EC2

- Amazon EC2 provides the following features:
- Virtual computing environments, known as *instances* Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits we need for our server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types.*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place.
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones.*

- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups.*

- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses.* Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources.

- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds*(VPCs)

AWS Elastic load balancer

Elastic Load Balancing distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of our application.
The load balancer serves as a single point of contact for us, which increases the availability of our application. We can add and remove instances from our load balancer as your needs change, without disrupting the overall flow of requests to our application. Elastic Load Balancing scales our load balancer as traffic to our application changes over time and can scale to the vast majority of workloads automatically. We can configure health checks, which are used to monitor the health of the registered instances so that the load balancer can send requests only to the healthy instances. We can also offload the work of encryption and decryption to our load balancer so that our instances can focus on their main work.
Elastic Load Balancing works with the following services to improve the availability and scalability of our applications.

- **Amazon EC2** — Virtual servers that run our applications in the cloud. We can configure our load balancer to route traffic to our EC2 instances.
- **Amazon ECS** — Enables us to run, stop, and manage Docker containers on a cluster of EC2 instances. We can configure our load balancer to route traffic to our containers. **Auto Scaling** — Ensures that we are running our desired number of instances, even if an instance fails, and enables we to automatically increase or decrease the number of instances as the demand on our instances changes. If we enable Auto Scaling with Elastic Load Balancing, instances that are launched by Auto Scaling are automatically registered with the load balancer, and instances that are terminated by Auto Scaling are automatically de-registered from the load balancer.
- **Amazon CloudWatch** — Enables us to monitor our load balancer and take action as needed.

- **Route 53** — Provides a reliable and cost-effective way to route visitors to websites by translating domain names (such as www.example.com) into the numeric IP addresses (such as 192.0.2.1) that computers use to connect to each other. AWS assigns URLs to our resources, such as load balancers.

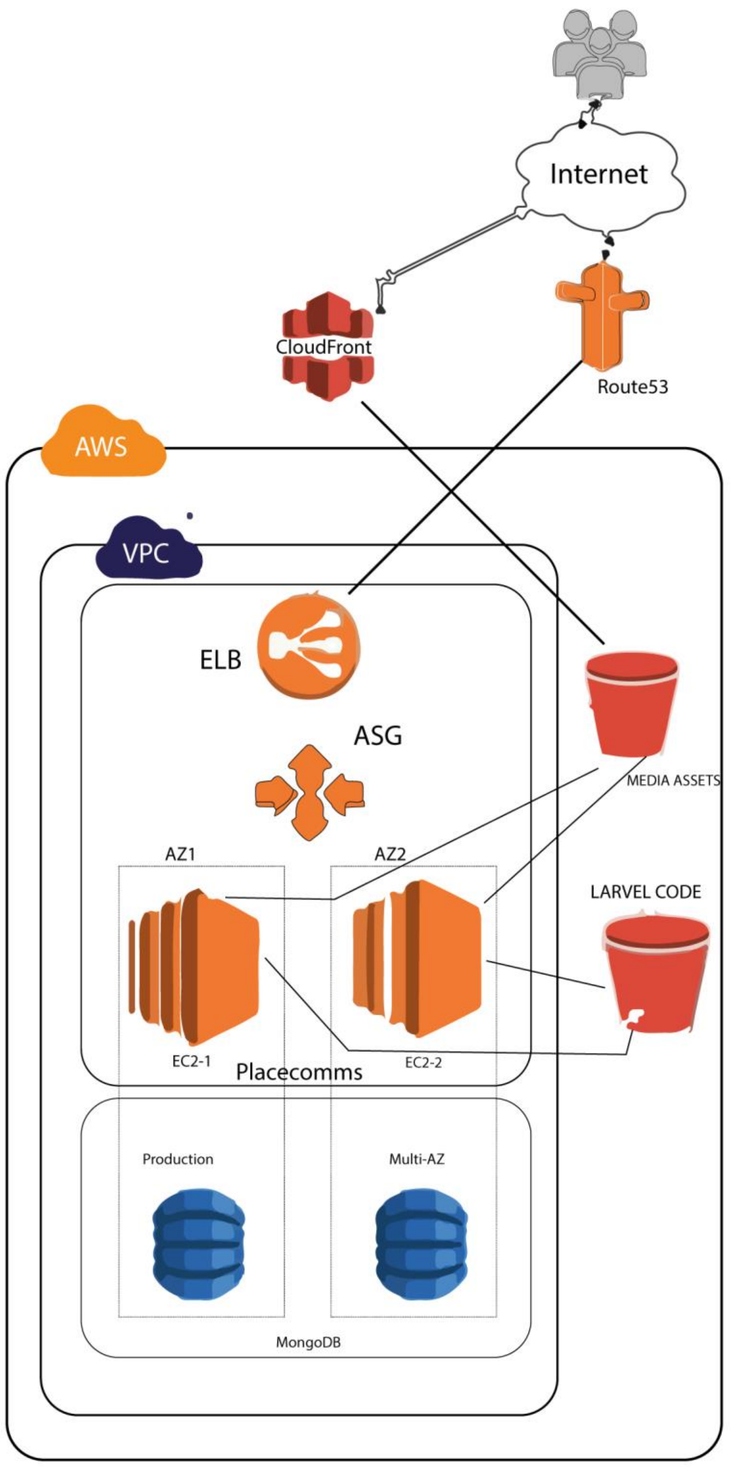
AWS S3

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. It is designed to deliver 99.999999999% durability, and stores data for millions of applications used by market leaders in every industry. S3 provides comprehensive security and compliance capabilities that meet even the most stringent regulatory requirements. It gives customers flexibility in the way they manage data for cost optimization, access control, and compliance. S3 provides query-in-place functionality, allowing you to run powerful analytics directly on your data at rest in S3. And Amazon S3 is the most supported cloud storage service available, with integration from the largest community of third-party solutions, systems integrator partners, and other AWS services.

1.3 Features:

1.3.1 Security

Security Flow Diagram



-

- **Data Protection**
  Data is organised and classified into segments such as publicly available, available to only members of the organisation or certain members of the organisation, available to custom audience.

- **Privilege management**
  We included a privilege access system using AWS IAM so that people are only able to access what they need.

- **Infrastructure Protection**
  Most importantly every data is encrypted whether is at rest or transit by using Encrypted EBS volumes in EC2 instances and S3 buckets along with 256 bit ssl encryption on website.Well configured VPC with NAT and firewall for the cloud infrastructure protection. Outside of cloud typically data center is protected by security rfid controls, cctv, lockable cabinets etc. Within our virtual data center these security is handled by the infrastructure protections we implemented at a VPC level.

- **Detective Controls**
  Use of Detective controls to detect or identify a security breach by the help of using AWS Cloudtrail. Storage of all system logs and Cloudtrail health check alarm is also used for extensive protection.

- **Reliability**
  Our framework possesses the ability to recover from service or infrastructure outages or disruptions as well as the ability to dynamically acquire computing resources to meet demand.The system has constant monitoring done by using AWS CloudWatch to detect any changes to the servers and react.We used CloudWatch to monitor the vpc and autoscaling to automate change in response to change in the production environment.

- **Performance Efficiency**

Compute

Since aws servers are virtualized and at the click of a button (or API Call) we can change the type of server in which our application is running.

Very high level of performance is achieved by using compute optimized general purpose T2.xlarge instance coupled with high frequency Intel Xeon Processors. Our servers can sustain high CPU performance for as long as workload is needed.

Storage

At aws storage is virtualized, by using aws S3 we have high data durability and, cross region replication etc. Encrypted SSD based EBS volumes are used in all of the instances for high IOPS speed.

Database

To overcome the relational databases approach and limitations MongoDB is used. With Mongodb we developed that were never possible with traditional relational database like fast, iterative development, flexible data model, distributed data platform, multi document acid transactions. Mongodb shared clusters with 3 replica sets with same t2, xlarge instances were used to ensure data redundancy and that the systems are highly available.

Space-time trade off

Using Aws RDS to add read replicas, reducing the load of the database and creating multiple copies of either database or main app instance to help in lowering latency.Taking advantage of the aws global infrastructure the vpc have multiple copies of the instances in region closest to the server. Caching services such as AWS ElastiCache and CloudFront is used to reduce latency further.

Cost Optimization

For excessive cost optimization all the resources are neither over provision nor under provision, instead it grows as the demand or traffic grows with autoscaling which scales on demand. CloudWatch is also used for keeping track of what the required resource should be also the use of allocation tags, billing alerts, or consolidated billing.

1.4. Specifications

- Server: AWS EC2
- CPU: High frequency Intel Xeon 16 core
- RAM: 16gb
- Storage: Amazon ebs SSD
- Website, HTML5, CSS3, JS, MongoDB, Laravel (PHP framework), all running on apache2, Ubuntu 16.04 LTS.
- Database: MongoDB shared cluster of 3 instances deployed on AWS ec2. With load balancing.
- Payment: Instamojo payment gateway
- Mass email: AWS ses, AWS sqs
- Transactional SMS- AWS SNS Misc. other jobs: AWS Lamba
- Website load Management: AWS elastic load balancer
- Hosting of static web content: AWS S3

Chapter 3 SYSTEM REQUIREMENT SPECIFICATION

3.1. Software Requirements

- Ubuntu 17.04 LTS
- Apache2 server
- PHP 7.2 /Laravel 5.6
- MongoDB 3.6

3.2. Hardware Requirements

Virtual Hardware

- AWS EC2
- t2.xlarge general purpose instance
- high frequency intel xeon 16 core
- Ram- 16gb
- Storage- amazon ebs ssd

Chapter 7 PROJECT PLANNING

7.1. PROJECT TIMELINE

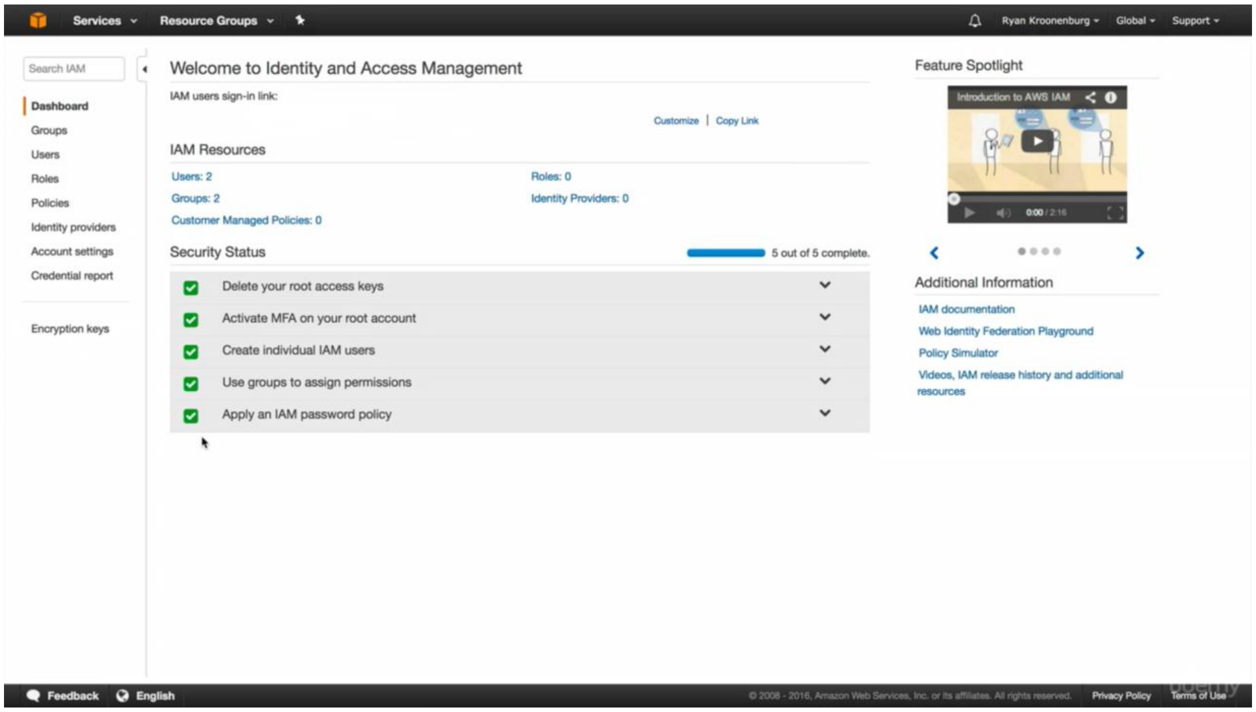
The project is said to be done in a time frame of 1 academic semesters. The detailed division of the project phases is as follows:

Project Idea	Feb 2018
Research	Feb 2018
Tools Gathering	Feb -March 2018
Model Designing	March 2018
Unit Testing Phase I	March 2018
Unit Testing Phase II & Final Testing	April 2018
Final Submission	30th April 2018

Chapter IMPLEMENTATION

8.1 User Access Control

[Screenshot]



Welcome to Identity and Access Management

IAM users sign-in link:

IAM Resources

Users: 2	Roles: 0
Groups: 2	Identity Providers: 0
Customer Managed Policies: 0	

Security Status — 5 out of 5 complete.

- ✓ Delete your root access keys
- ✓ Activate MFA on your root account
- ✓ Create individual IAM users
- ✓ Use groups to assign permissions
- ✓ Apply an IAM password policy

Additional Information

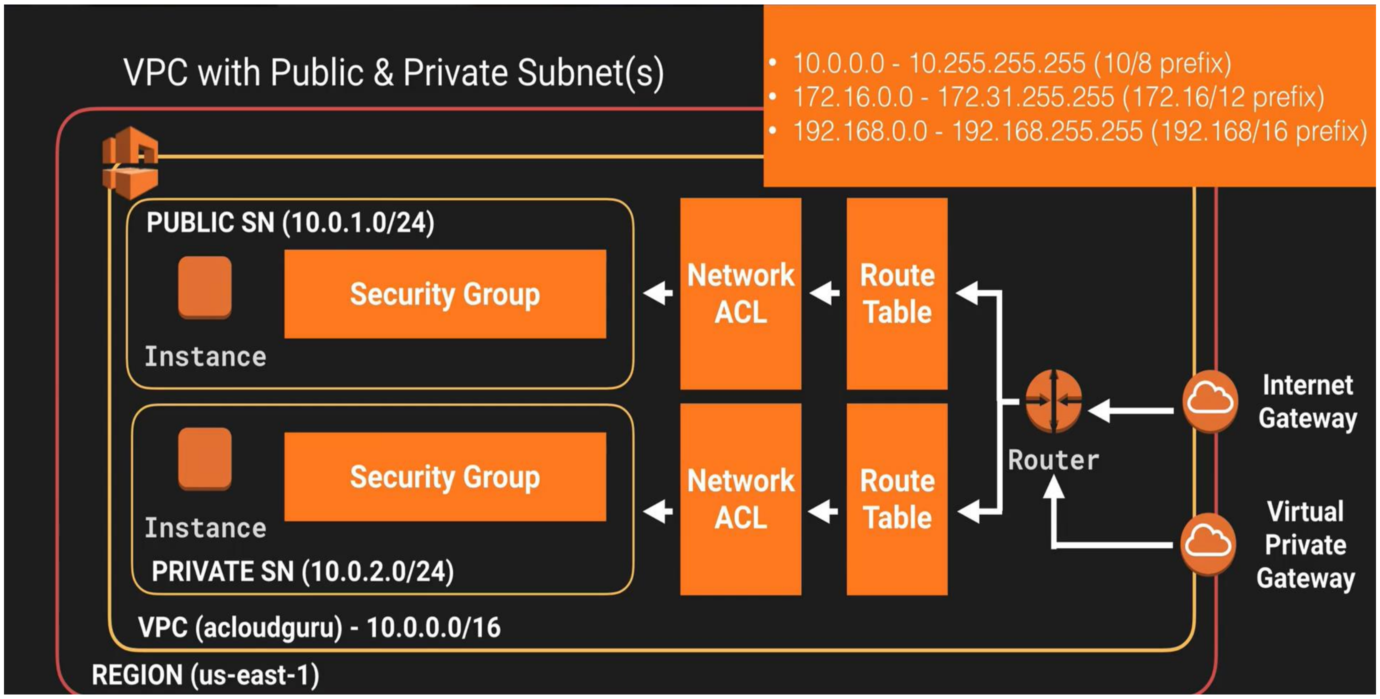
- IAM documentation
- Web Identity Federation Playground
- Policy Simulator
- Videos, IAM release history and additional resources

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

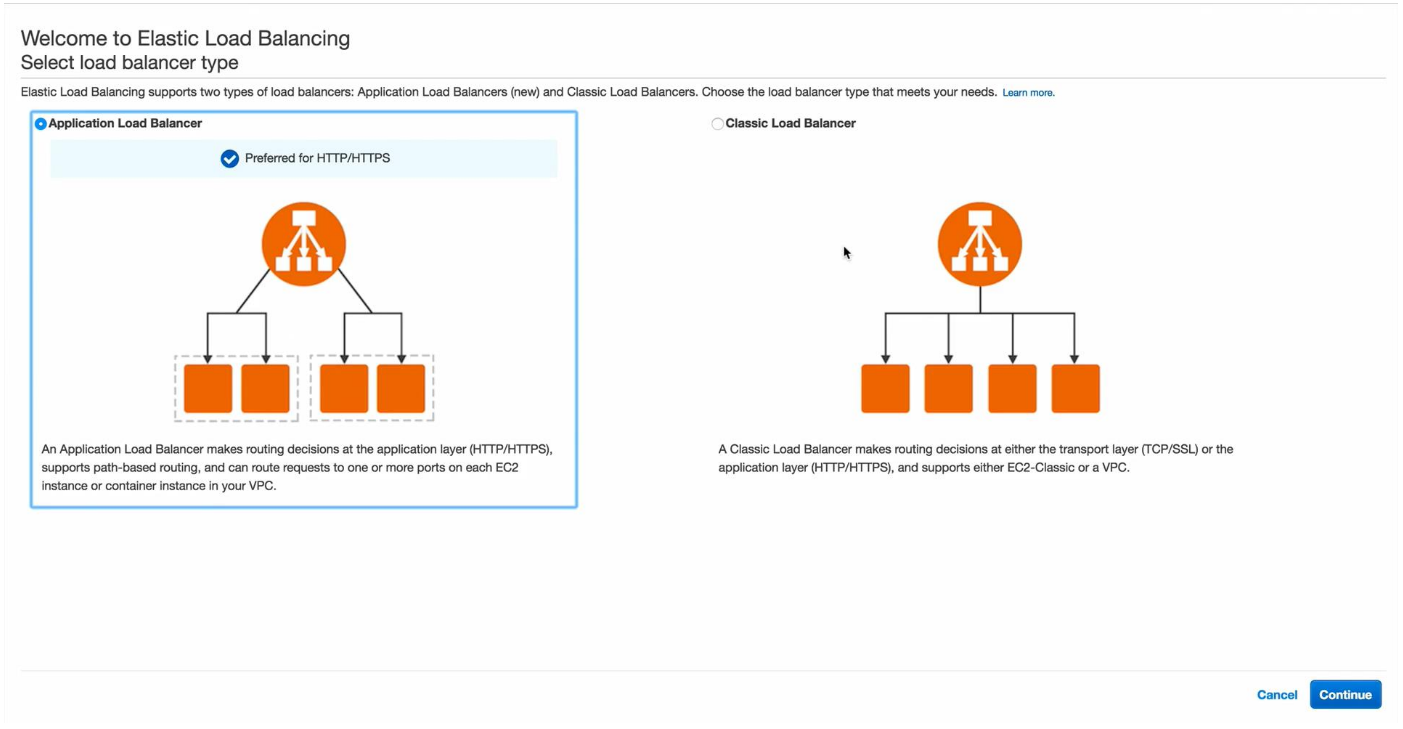
For better security we have to create user and group using AWS Identity Access Management with appropriate access to the aws services. Also enabling multi factor authentication in the AWS IAM control panel will improve security more safeguarding from hackers. By creating specifying user and roles by defining their privileges xml file we can ensure that only operate

8.2 Virtual Private Cloud

AWS VPC is the backbone of our framework enabling us to deploy and manage our our custom define private network of servers which can communicate with each other as well as the internet with proper firewall NAT instance.
We had to create a public facing subnet for our servers that have access to the internet, and place our backend systems such as mongodb database and application servers in a private facing subnet with no internet access.
We had leveraged multiple layer of security including security groups and network access control lists, to help control access to our server instances in each subnet.


VPC with Public & Private Subnet(s)
• 10.0.0.0 - 10.255.255.255 (10/8 prefix)
• 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
• 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
PUBLIC SN (10.0.1.0/24)
Security Group
Instance
Network ACL
Route Table
Internet Gateway
Router
Security Group
Instance
PRIVATE SN (10.0.2.0/24)
Network ACL
Route Table
Virtual Private Gateway
VPC (acloudguru) - 10.0.0.0/16
REGION (us-east-1)

8.3 Load Balancer and Auto Scaling



Welcome to Elastic Load Balancing

Select load balancer type

Elastic Load Balancing supports two types of load balancers: Application Load Balancers (new) and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more.

[✓ Application Load Balancer] Classic Load Balancer

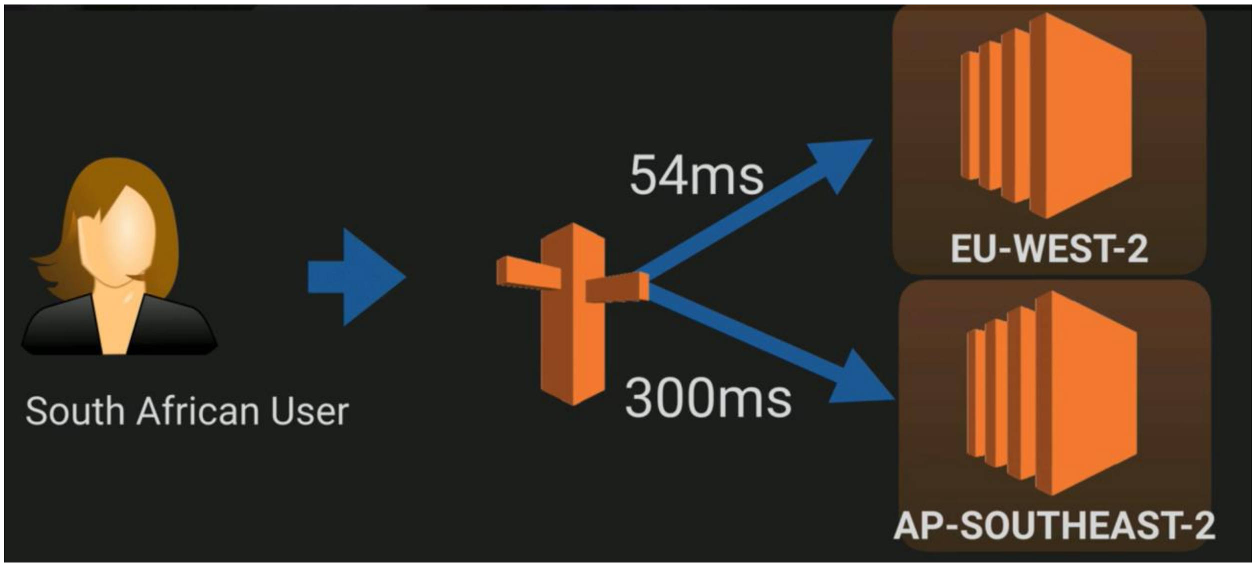
Preferred for HTTP/HTTPS

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.

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.

In case of heavy traffic surge to the website we decide to use the AWS Elastic Load Balancer and auto scaling service to automatically launch a new instance in case of server failure or heavy traffic.
For this process we had to associate an elastic ip address to each of our instances. We implemented Application Load balancer instead of classic load balancer to enhance the performance.
Next we created a custom auto scaling group with scaling configuration between 1 to 3 instances, it means no of instance will automatically be increased or decreased based on the load.

8.4 Custom user DNS routing



One of the most important feature of our framework is the capability to auto redirect user’s request to the nearest available server to reduce network latency and server response time.
To use the latency based routing efficiently we created a latency resource record set for the ec2 instances in each region that host our application.
When Route 53 receives a query for our system, it selects the latency resource record set for the region that gives the user the lowest latency. Route 53 then responds to the value associated with that resource record set which are backend instances and mongodb instances in our case.

8.5 Installing Server Applications

Screenshot

Our apache servers were configure on t2 AWS ec2 instances running on Ubuntu 17.04 LTS Ami, which is know for its robust server performance.

Through the ec2 terminal we had to setup apache server, php 7.2, laravel framework along with all necessary dependents plugins like curl, zip, aws sdk etc.

Chapter 10 CONCLUSIONS AND FUTURE SCOPE

10.1. Final Remarks

This "Placecomms: Smart Online Recruitment Platform" project can solve the employment crisis happening in Indian colleges.

It can boost up the employment rate and help facilitate the placement recruitments program to be executed smoothly.

10.3. Future Scope

- In future many more modules can be added to better automate our recruitment process.
- We can apply Data Analytics concepts, to analyse and emphasize proper power and electricity consumption measures for the cloud architecture.
- Applying real time analytics to determine employment stats.