CSYE6225

NETWORK STRUCTURE & CLOUD COMPUTING

"CLOUD9"

DECEMBER 2016

Guided By:

Prof. Julian Cooper

By:

Achal Sancheti
Bhavesh Patel
Saurabh Chaudhari
Srilaya Yallapragada
Sumit Deshmukh

Table of Contents:

INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort.

'Cloud' refers to a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources. The term cloud is originated as a metaphor for the Internet. It is important to note the differences between the term 'cloud' and cloud symbol from Internet.

Cloud computing typically provides 3 types of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). These services are available over the Internet to the whole world where the cloud acts as a single point of access for serving all the customers.

Types of Cloud

- Private Cloud – A private cloud is a particular model of cloud computing that involves a distinct and secure cloud based environment in which only the specified client can operate.
- Public Cloud – The public cloud is defined as a multi-tenant environment, where you buy a "server slice" in a cloud computing environment that is shared with a number of other clients or tenants.
- Hybrid Cloud – Hybrid cloud is a cloud computing environment which uses a mix of on-premises, private cloud and third-party, public cloud services with orchestration between the two platforms.
- Community Cloud – It is similar to public cloud except that its access is limited to a specific community of cloud consumers.

REQUIREMENT

- Develop a login portal using Java Spring which will consist of use-case buttons to simulate an increasing load on the application and database.
- The database can either be a relational database or a NoSQL database.
- The entire stack must be deployed in AWS or Azure.
- There are 5 use-case buttons that need to be implemented which are follows:
  - Simulation of 3 user login to application and run 2 report going back 10 days
  - Simulation of 10 user login to application and run 6 report going back 30 days
  - Simulation of 17 user login to application and run 10 report going back 60 days
  - Simulation of 24 user login to application and run 14 report going back 90 days.
  - User input fields to allow me the ability to enter parameters for how many user's login simulation and how many reports those users are uniquely being running.
- Proper infrastructure alerts and triggers to allow for auto-scaling of resources to accommodate the additional load in application, network, data storage and usage with your environment.
- Minimum of one load balancer is required.

BUSINESS JUSTIFICATION

To deploy a spring MVC web application in AWS with the help of MYSQL RDS database to scale up/down the server according to user traffic and hence avoiding latency and bottlenecks.

SYSTEMS AND DESCRIPTIONS

Platform

Typically, when we need to choose a cloud based platform then only two of them comes in the mind – Microsoft's Azure or Amazon's AWS.

When we started to research on these two platforms, we observed following comparisons:

- There are 38 instance types on AWS whereas 33 instance types on Azure.
- There are 7 instance families on AWS whereas 4 instance families on Azure.
- Zones are available on AWS.
- For DNS AWS uses Route53.

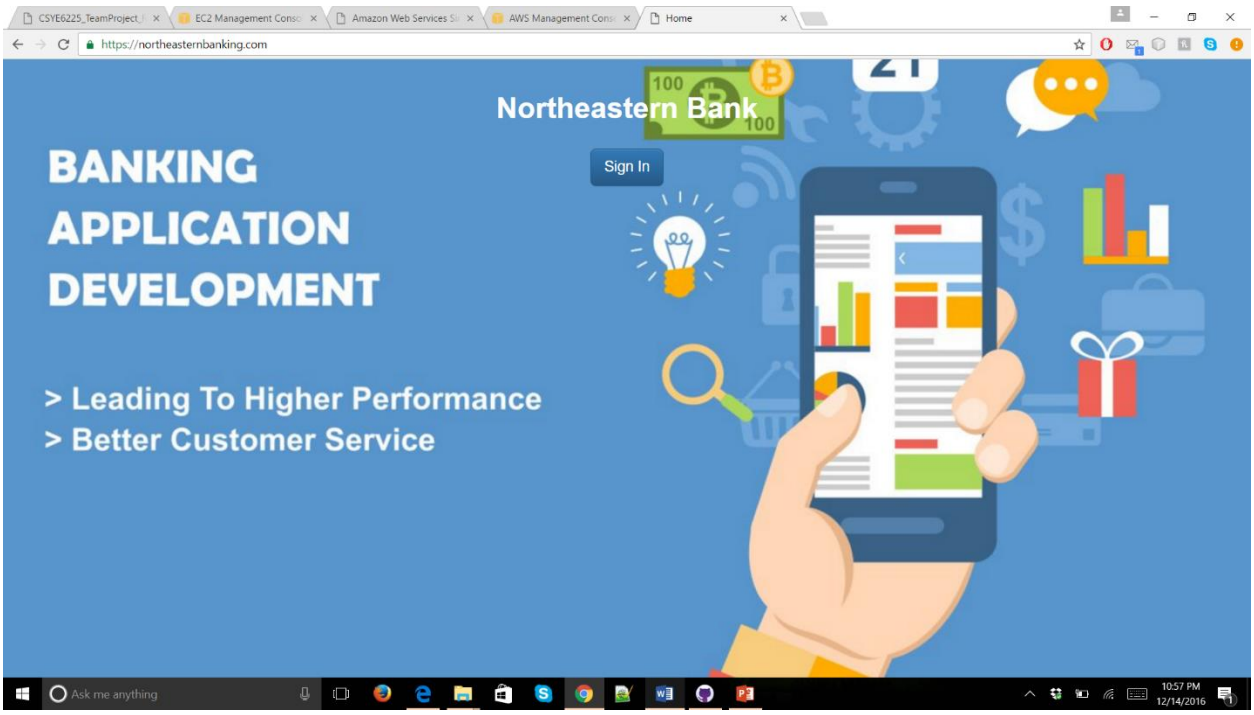
Apart from the above comparisons, we also felt that the performance speed of Azure is little slow as compared to AWS, as we have used both of them previously. So we decided to build our entire project on Amazon Web Services.

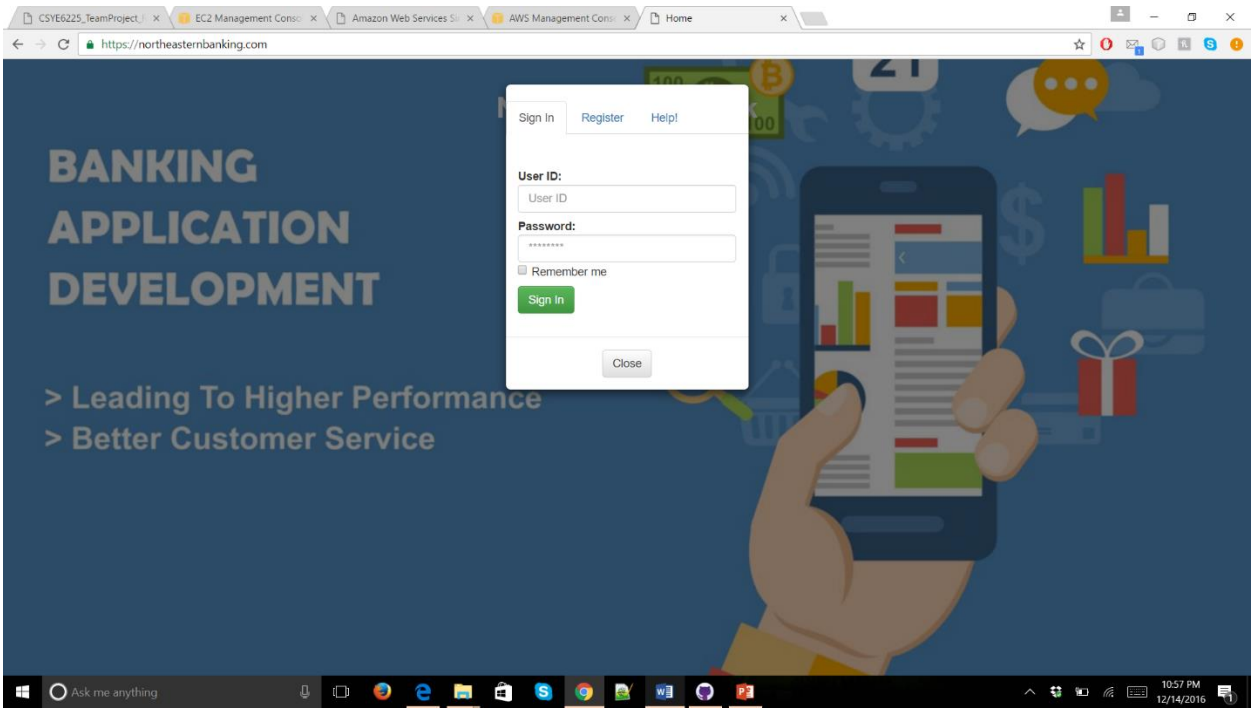
Techniques Used

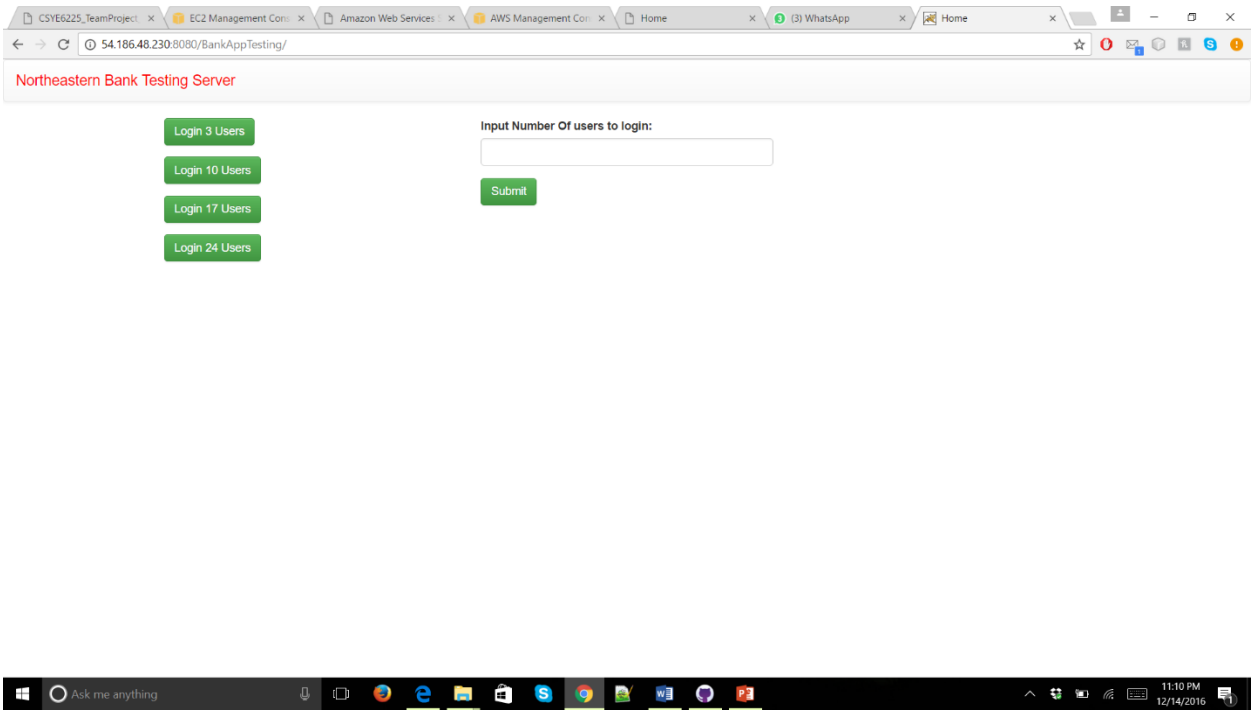
- Elastic Load Balancing
- Auto Scaling
- Cloud Watch
- Health Check

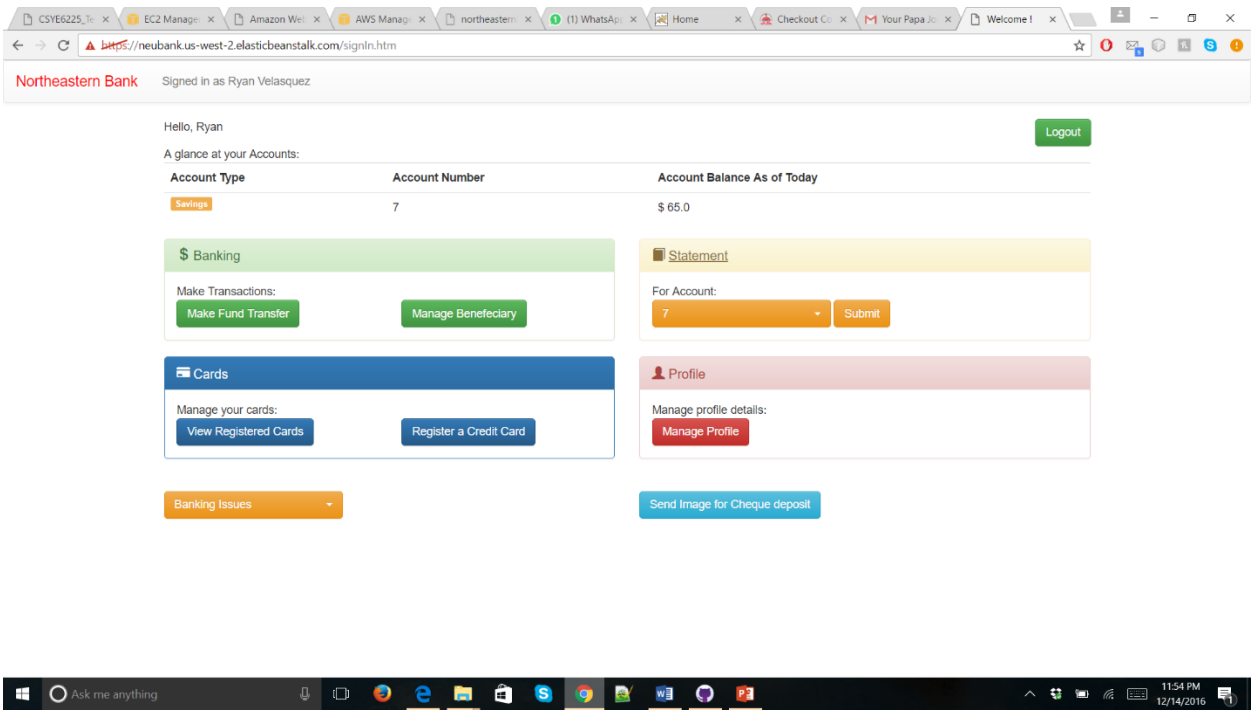
Web Application

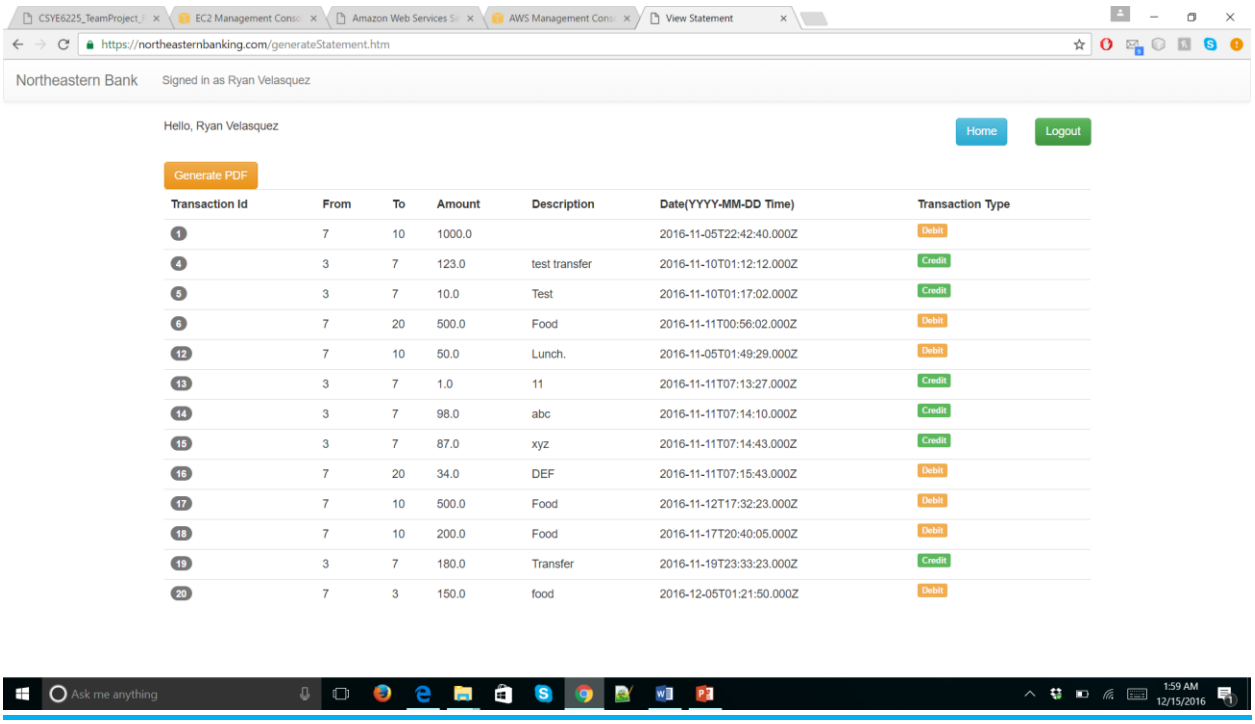
- We have created a banking application using Spring MVC framework, whose link is https://northeasternbanking.com/
- We are using MySQL with RDS as a database.
- The application has a login mechanism and also a report page to view the report of each transaction.
- We have written a Java code to register new user on click of use case buttons. We have stored the information of the created users in the session, that we are using on next page.


https://northeasternbanking.com
Northeastern Bank
Sign In
BANKING
APPLICATION
DEVELOPMENT
> Leading To Higher Performance
> Better Customer Service


https://northeasternbanking.com
Sign In
Register
Help!
User ID:
User ID
Password:
Remember me
Sign In
Close
BANKING
APPLICATION
DEVELOPMENT
> Leading To Higher Performance
> Better Customer Service

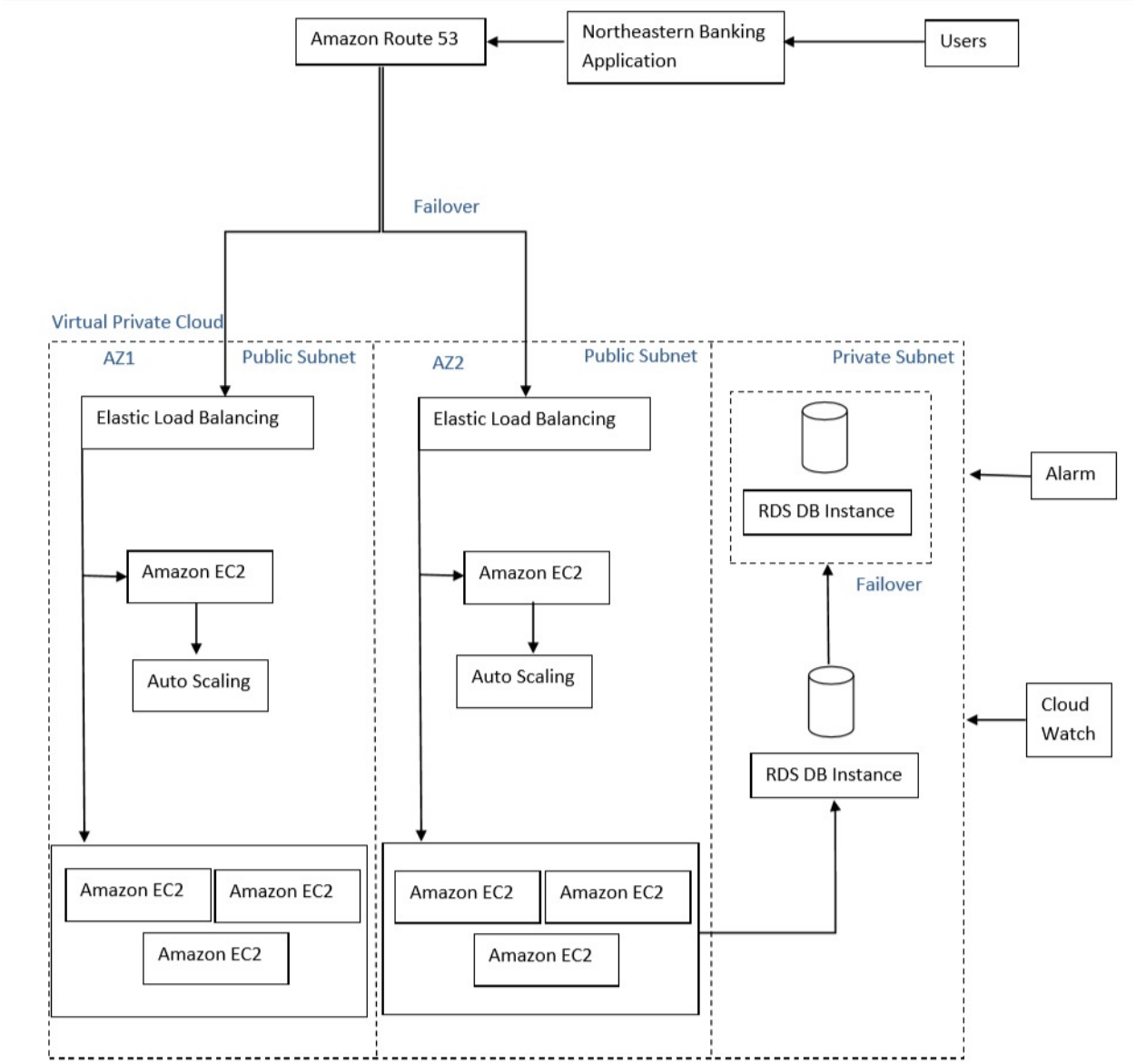

54.186.48.230:8080/BankAppTesting/
Northeastern Bank Testing Server
Login 3 Users
Login 10 Users
Login 17 Users
Login 24 Users
Input Number Of users to login:
Submit
Ask me anything
11:10 PM
12/14/2016


https://neubank.us-west-2.elasticbeanstalk.com/signIn.htm
Northeastern Bank
Signed in as Ryan Velasquez
Hello, Ryan
Logout
A glance at your Accounts:
Account Type
Account Number
Account Balance As of Today
Savings
7
$ 65.0
$ Banking
Make Transactions:
Make Fund Transfer
Manage Benefeciary
Statement
For Account:
7
Submit
Cards
Manage your cards:
View Registered Cards
Register a Credit Card
Profile
Manage profile details:
Manage Profile
Banking Issues
Send Image for Cheque deposit
Ask me anything
11:54 PM
12/14/2016



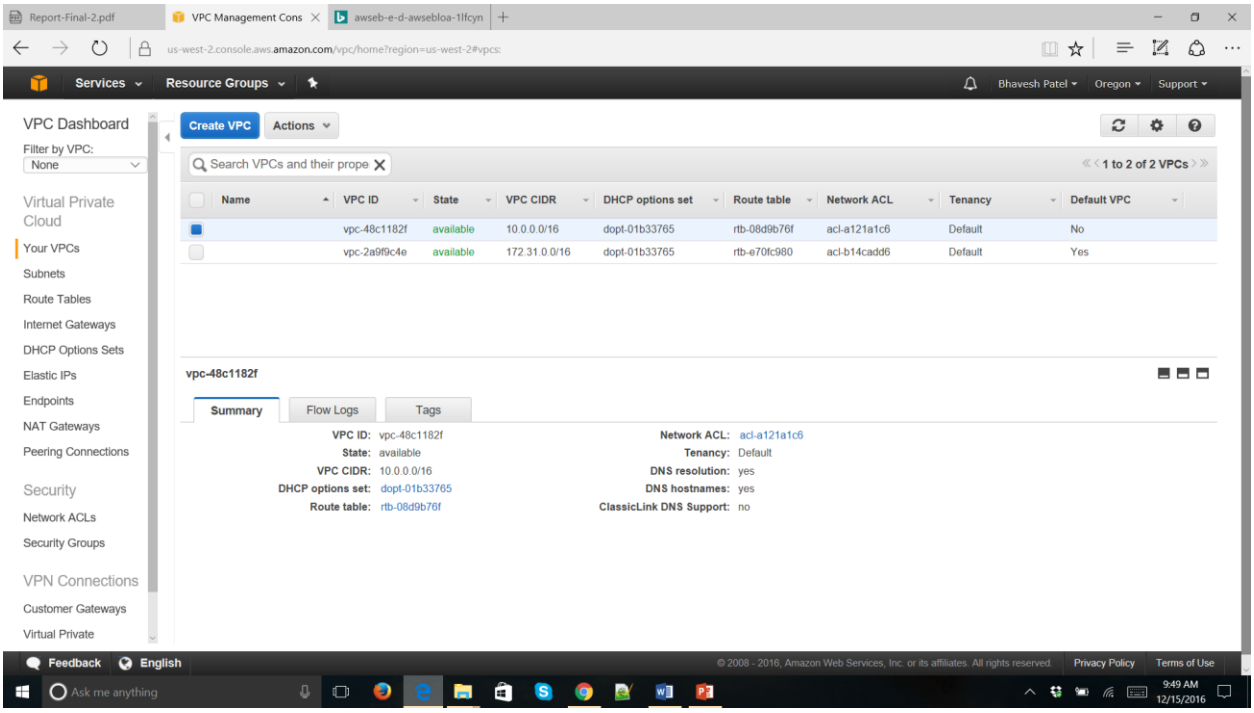
CLOUD ARCHITECTURE

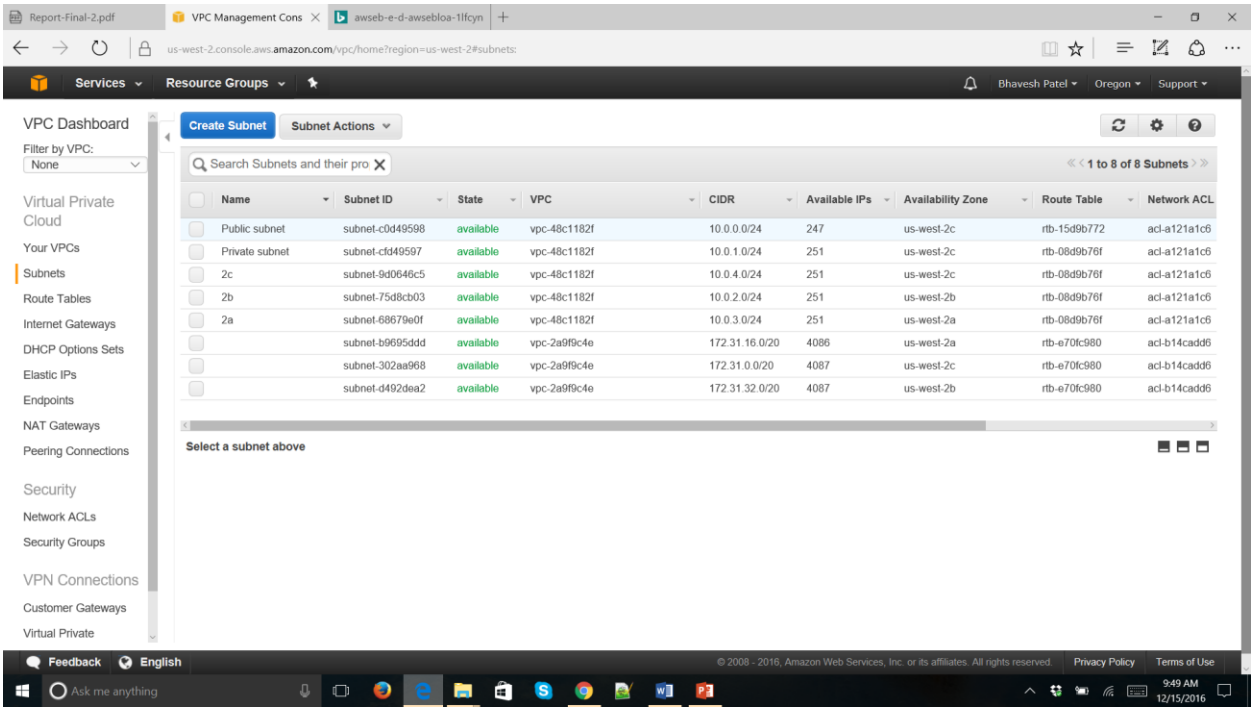
Visio Diagram



Virtual Private Cloud

- Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define.
- We have created a Virtual Private Cloud with Public and Private subnets.
- This VPC consist of EC2 instance, Load Balancer, Auto Scaling Group and RDS instance.



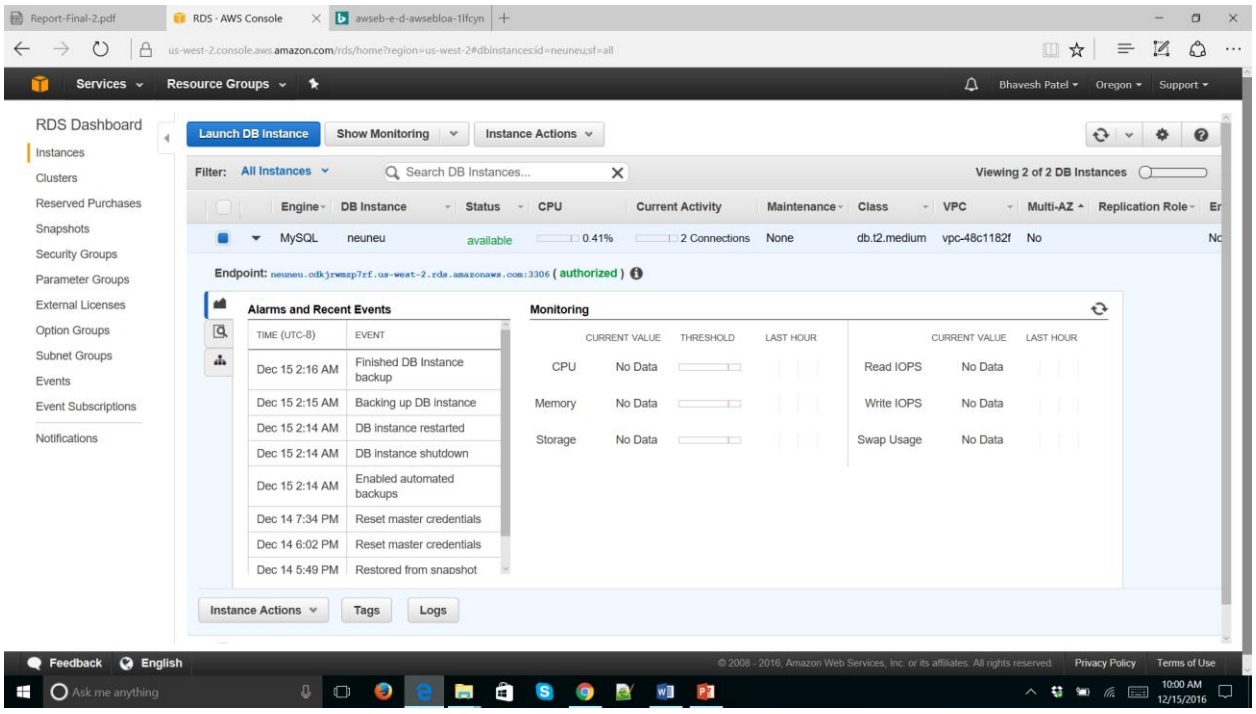


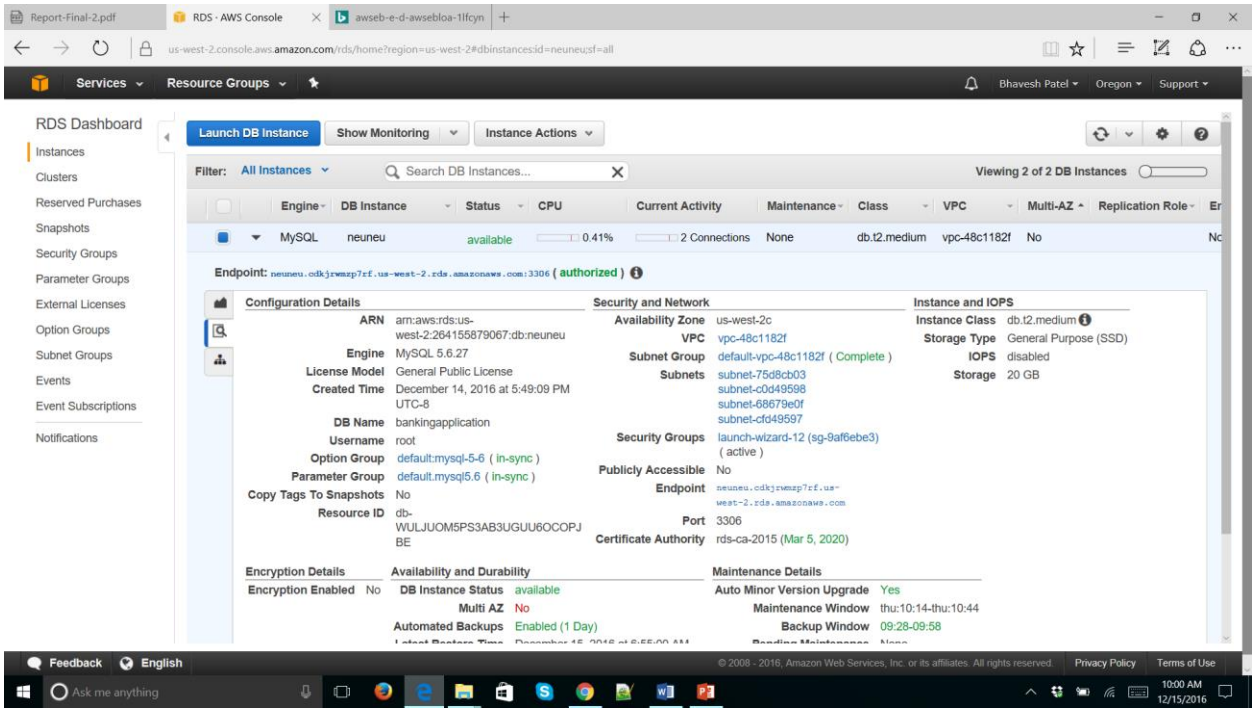
Amazon RDS for MySQL

- **Amazon** Relational Database Service (**Amazon RDS**) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.

It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

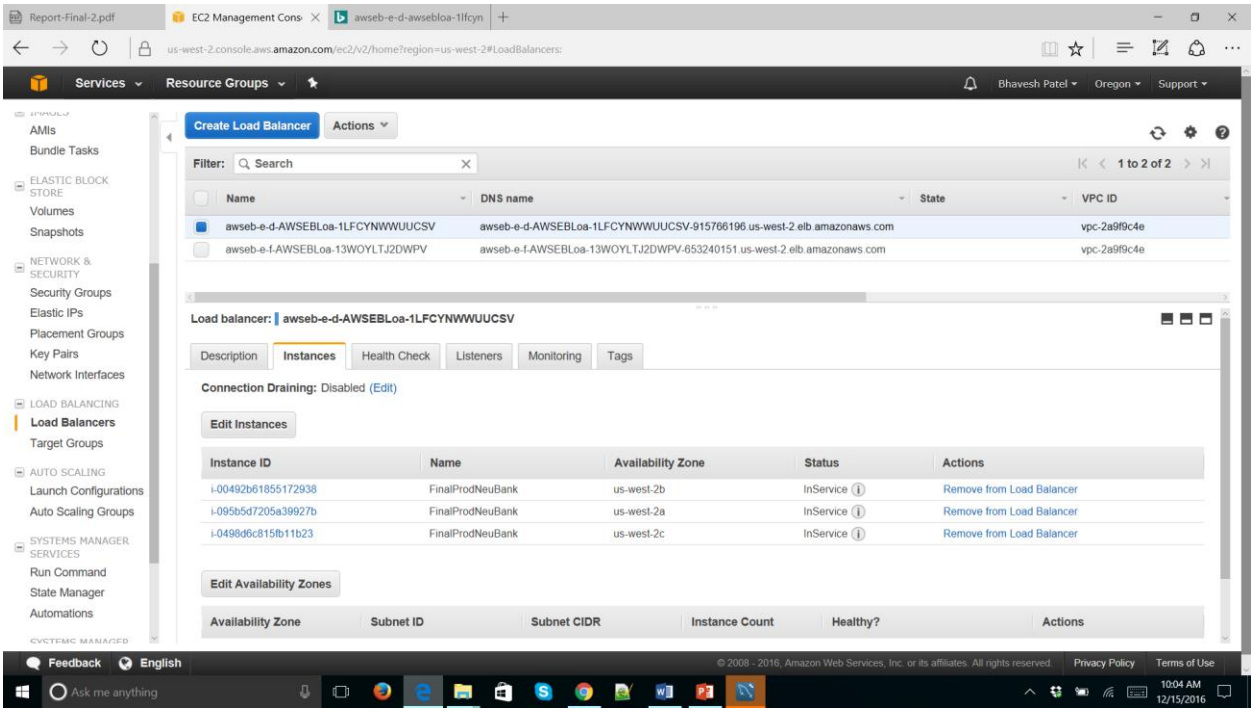
- In order to secure the data, we have created the RDS instance in the private subnet of Virtual Private Cloud.
- For the database we have selected MySQL, with db.t2.medium as DB instance class.
- The DB instance is connected to MySQL Workbench, that is protected with a username and password.





Elastic Load Balancer

- Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications, seamlessly providing the required amount of load balancing capacity needed to route application traffic.
- Benefits:
    - Available
    - Elastic
    - Secure
- We have applied SSL certificate to make our application more secure. We have attached it in the load balancer so that HTTPS request is forwarded to our EC2 instance hosting the application. We have created the add listener with:
    - LB Protocol – HTTPS
    - LB Port – 443
    - Instance Protocol – HTTPS



Web Application Security

- We used certificate manger service from AWS to get a secure signed certificate
- We have secured our database by putting it inside the VPC private network and dismissing its access from the internet
- We are redirecting HTTP traffic to HTTPS and forming a SSL connection with all our clients

Auto Scaling

- Auto Scaling allows us to scale our Amazon EC2 capacity up or down automatically as per conditions we define.
- We are using 3 dedicated t2 micro instances to support our usual load and configured a launch configuration with Tomcat and Java preinstalled.
- The launch configuration has a security group that will define the traffic on each of our listening port
- We have at max 5 instances to scale up to and scaling policies as below:
  Scale Up: 1) On CPU Utilization > 80% for 5 minutes

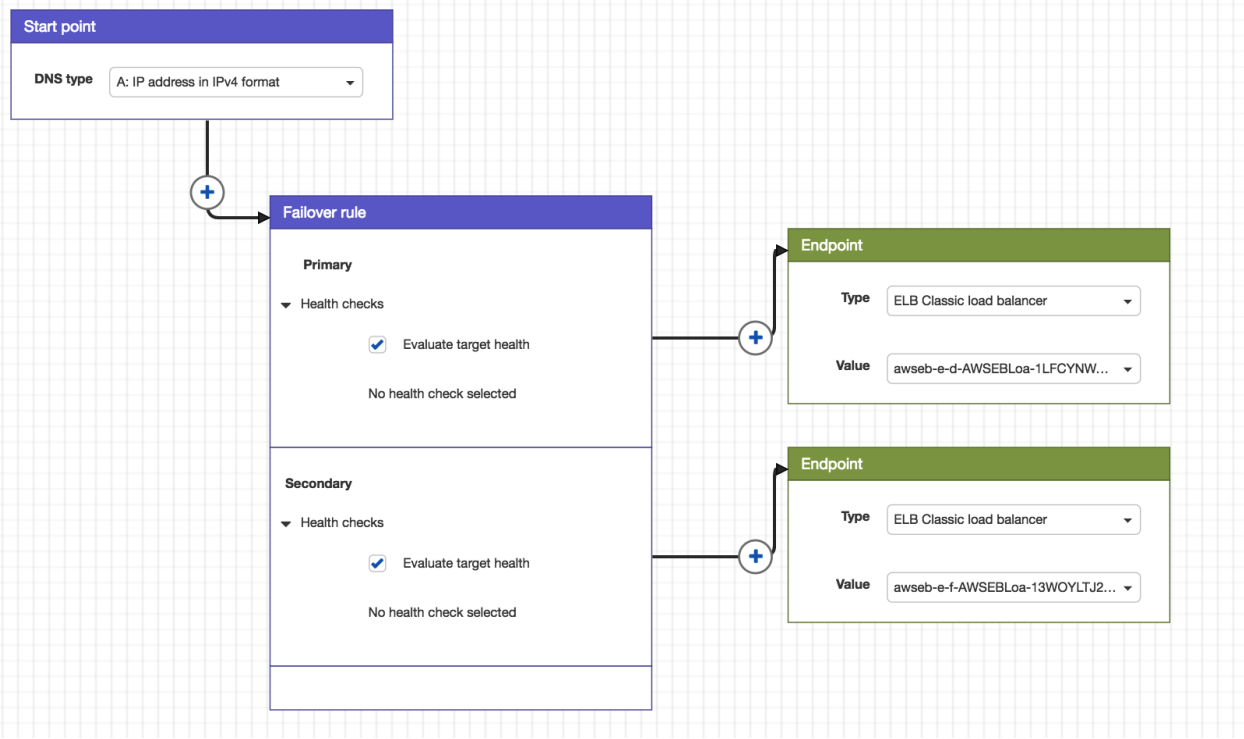
2) On Network In > 60 MB in 1 minute Interval

Scale Down: 1) On CPU Utilization < 10% for 5 minutes

2) On Network In < 20 MB in 1 minute Interval

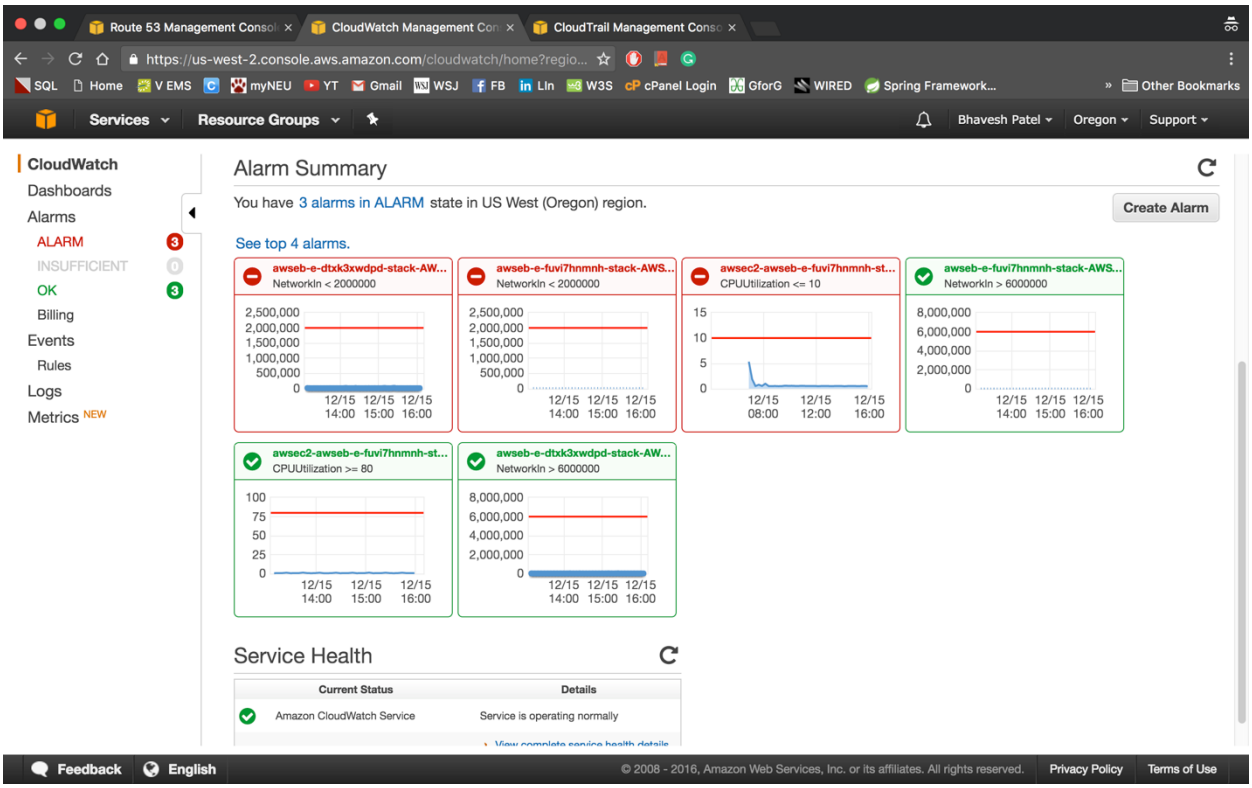
Route53

- This service allows us to tie our Certificate to the domain northeasternbanking.com
- This also allows us to switch to a different Infrastructure in case of failure.



Cloud Watch

- Amazon CloudWatch is a component of Amazon Web Services (AWS) that provides monitoring for AWS resources such as Amazon EC2 instances, Load Balancers and Amazon RDS instances and the customer applications running on the Amazon infrastructure.
- We have set the Alarms in Auto Scaling group to monitor the performance of our application.
- We have one alarm for CPU Utilization greater that 80% for 1 minute when we will scale up the infrastructure by 1 instance.
- An alarm can have three possible states:
    - OK—The metric is within the defined threshold
    - ALARM—The metric is outside of the defined threshold
    - INSUFFICIENT_DATA—The alarm has just started, the metric is not available, or not enough data is available for the metric to determine the alarm state

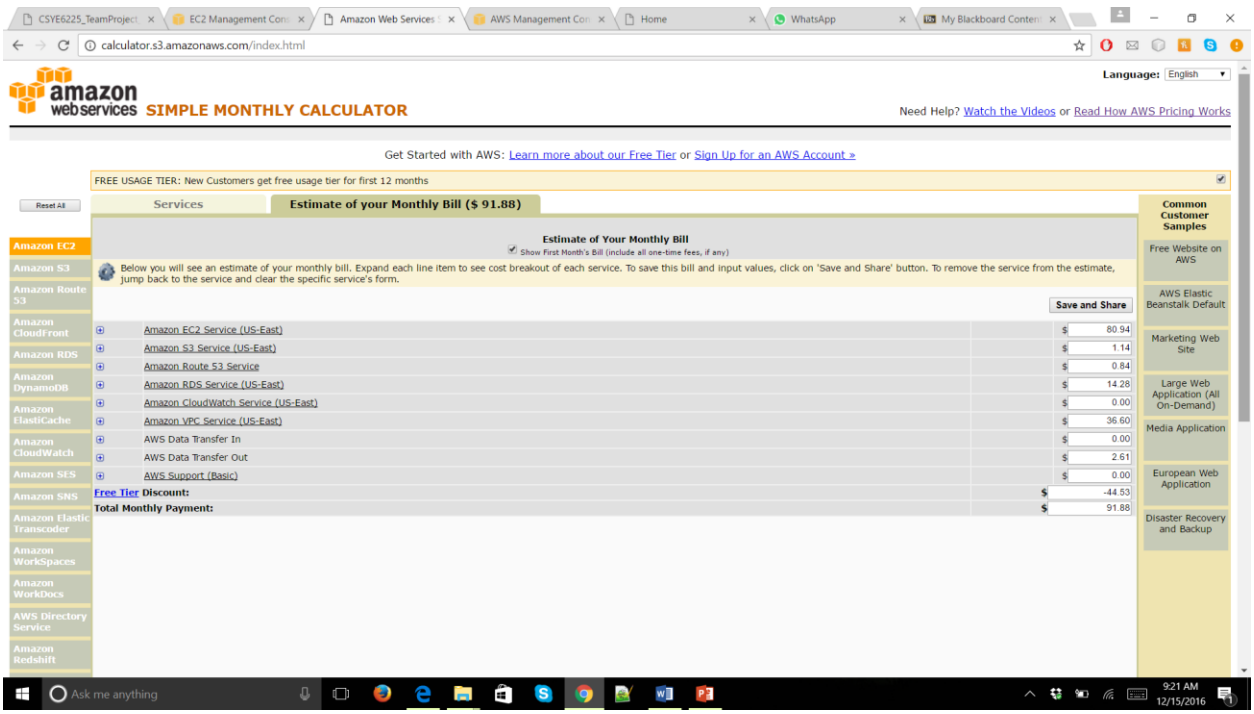

Route 53 Management Consol
CloudWatch Management Con
CloudTrail Management Conso
https://us-west-2.console.aws.amazon.com/cloudwatch/home?regio...
Services
Resource Groups
Bhavesh Patel
Oregon
Support
CloudWatch
Dashboards
Alarms
ALARM 3
INSUFFICIENT 0
OK 3
Billing
Events
Rules
Logs
Metrics NEW
Alarm Summary
You have 3 alarms in ALARM state in US West (Oregon) region.
Create Alarm
See top 4 alarms.
awseb-e-dtxk3xwdpd-stack-AW...
NetworkIn < 2000000
awseb-e-fuvi7hnmnh-stack-AWS...
NetworkIn < 2000000
awsec2-awseb-e-fuvi7hnmnh-st...
CPUUtilization <= 10
awseb-e-fuvi7hnmnh-stack-AWS...
NetworkIn > 6000000
awsec2-awseb-e-fuvi7hnmnh-st...
CPUUtilization >= 80
awseb-e-dtxk3xwdpd-stack-AW...
NetworkIn > 6000000
Service Health
Current Status
Details
Amazon CloudWatch Service
Service is operating normally
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Cloud Pricing Model for the next 3-6 months

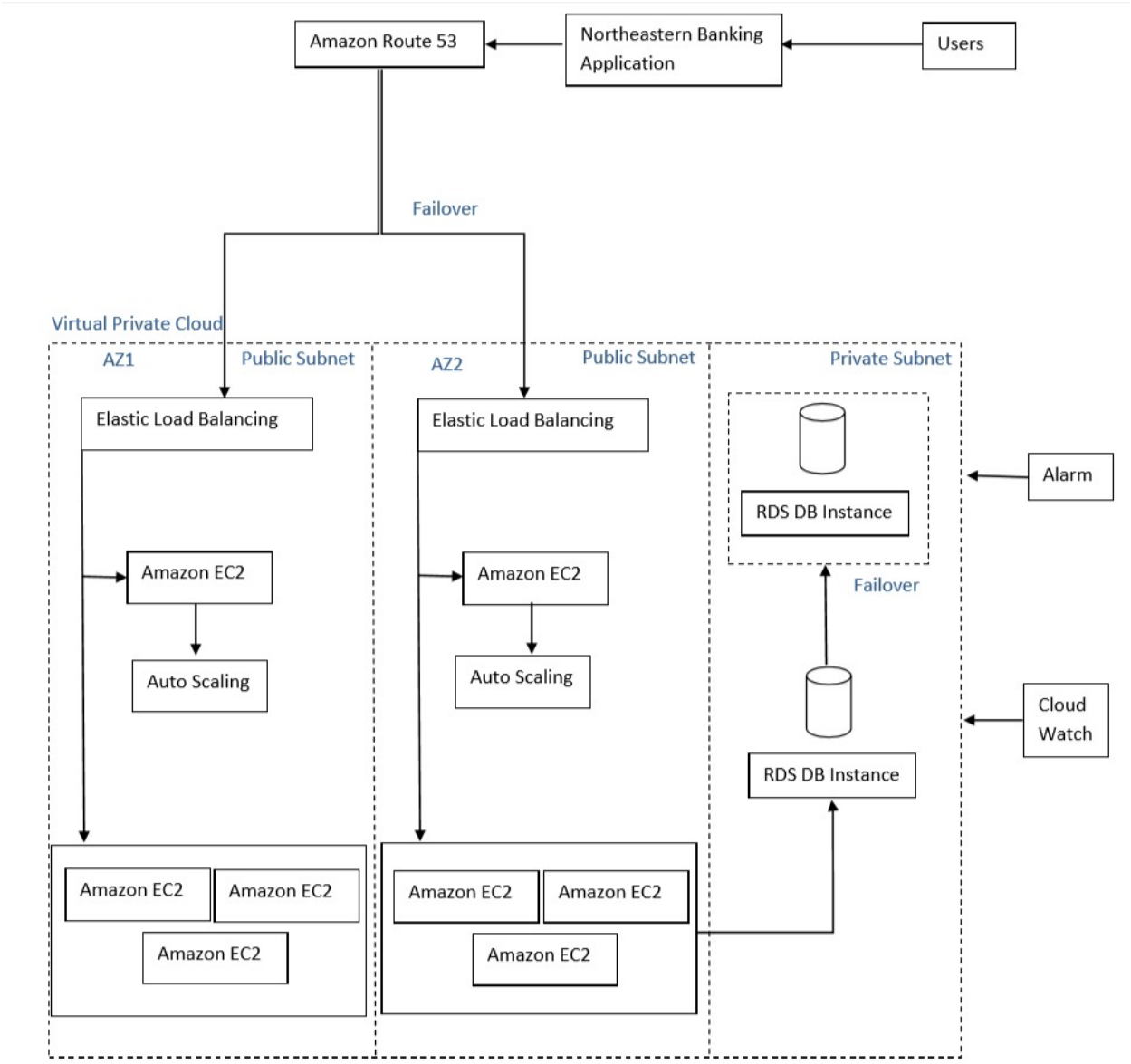
It turns out that the estimate cost of Northeastern Banking Application is $91.88 per month using the AWS Simple Monthly Calculator. So, for 6 months the cost will be $551.28.

The cost will also include $50 for each month, for route 53 and $12 to purchase a domain first year.

So, the overall cost for our application, for 6 months, would come around $857.28



DISASTER RECOVERY PROCEDURES



- Our cloud infrastructure has Disaster recovery plan in place. In this process if a primary network or a dedicate server in a zone or a zone in a whole collapses or crashes, there should be a procedure in which we try to bring in the whole consistent system.

- When primary system is down we configure route 53 to use the failover branch of the system. The entire load is redirected to this secondary system in such case.
- Database replica is already made in our system to utilize in such scenarios.

Steps for Disaster recovery plan:

1. First we take the entire backup of our DB twice a week and save it as snapshots in S3 bucket.
2. Then we would take incremental backup daily and store it into a different S3 bucket.

REFERENCES:

- https://aws.amazon.com/documentation/
- http://whatiscloud.com/
- http://searchcloudcomputing.techtarget.com/definition/cloud-computing