

## PROJECT REPORT ON

# WAKE n' BAKE

CSYE6225 NETWORK STRUCTURE & CLOUD COMPUTING

### Guided By:

Prof. Julian Cooper  
Northeastern University  
April 2017

### Submitted by:

Mansi Jain  
Piyush Saxena  
Riddhi Chouhan  
Swarna Dommeti  
Vivek Dalal

## Table of Contents:

<b>INTRODUCTION .....</b>	<b>3</b>
<b>Cloud based delivery .....</b>	<b>3</b>
<b>OBJECTIVE .....</b>	<b>5</b>
<b>SYSTEMS AND DESCRIPTIONS .....</b>	<b>5</b>
<b>Techniques Used .....</b>	<b>5</b>
<b>Web Application .....</b>	<b>7</b>
<b>CLOUD ARCHITECTURE .....</b>	<b>11</b>
<b>Route 53 .....</b>	<b>12</b>
<b>Internet Gateway .....</b>	<b>13</b>
<b>Elastic Load Balancer .....</b>	<b>14</b>
<b>NAT Gateway .....</b>	<b>15</b>
<b>Auto Scaling .....</b>	<b>16</b>
<b>Database .....</b>	<b>17</b>
<b>Single Queue System.....</b>	<b>18</b>
<b>CLOUD PRICING MODEL FOR THE NEXT 3-6 MONTHS.....</b>	<b>19</b>
<b>KEY POINTS .....</b>	<b>19</b>
<b>CHALLENGES.....</b>	<b>20</b>
<b>REFERENCES .....</b>	<b>21</b>

## INTRODUCTION

---

Cloud computing architecture refers to the components and subcomponents required for cloud computing. These components typically consist of a front-end platform (fat client, thin client, mobile device), back end platforms (servers, storage), a cloud based delivery, and a network (Internet, Intranet, Intercloud). Combined, these components make up cloud computing architecture. 'Cloud' refers to a distinct IT environment that is designed for 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.

### Cloud based delivery

---

- **Software as a service (SaaS)**

The software-as-a-**service** (SaaS) service-model involves the cloud provider installing and maintaining software in the cloud and users running the software from their cloud clients over the Internet (or Intranet). The users' client machines require no installation of any application-specific software - cloud applications run on the server (in the cloud).

- **Development as a service (DaaS)**

Data as a service is web based design construct where by cloud data is accessed through some defined API layer. DaaS services are often considered as a specialized subset of a Software as a service offering.

- **Platform as a service (PaaS)**

Platform as a service is cloud computing service which provides the users with application platforms and databases as a service.[3] This is equivalent to middleware in the traditional (non-cloud computing) delivery of application platforms and databases.[6] We can take an example for this as Microsoft Azure provides platform as services for multiple languages, if we use .net platform then we can build products using .net framework which will be provided by Microsoft Azure.

- **Infrastructure as a service (IaaS)**

Infrastructure as a service is taking the physical hardware and going completely virtual (e.g. all servers, networks, storage, and system management all existing in the cloud). This is the equivalent to infrastructure and hardware in the traditional (non-cloud computing) method running in the cloud. In other words, businesses pay a fee (monthly or annually) to run virtual servers, networks, storage from the cloud. This will mitigate the need for a data center, heating, cooling, and maintaining hardware at the local level

## OBJECTIVE

---

- Develop a login portal using Java Spring to simulate an increasing load on the application and database with Apache JMeter.
- The database can either be a relational database or a NoSQL database.
- The entire stack must be deployed on AWS.
- Proper infrastructure alerts and triggers to allows auto-scaling of resources to accommodate the additional load in application, network, data storage and usage with your environment.

## SYSTEMS AND DESCRIPTIONS

---

- Platform

We have chosen Amazon's AWS platform to support our functionalities and architecture. Amazon Web Services (AWS) is a subsidiary of Amazon.com that offers on-demand cloud computing platforms. These services operate from 16 geographical regions across the world. They include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3".

## Techniques Used

---

- Database
  - RDS
  - Aurora Replica
- Web Application
  - Spring
  - Hibernate
  - Spring Security

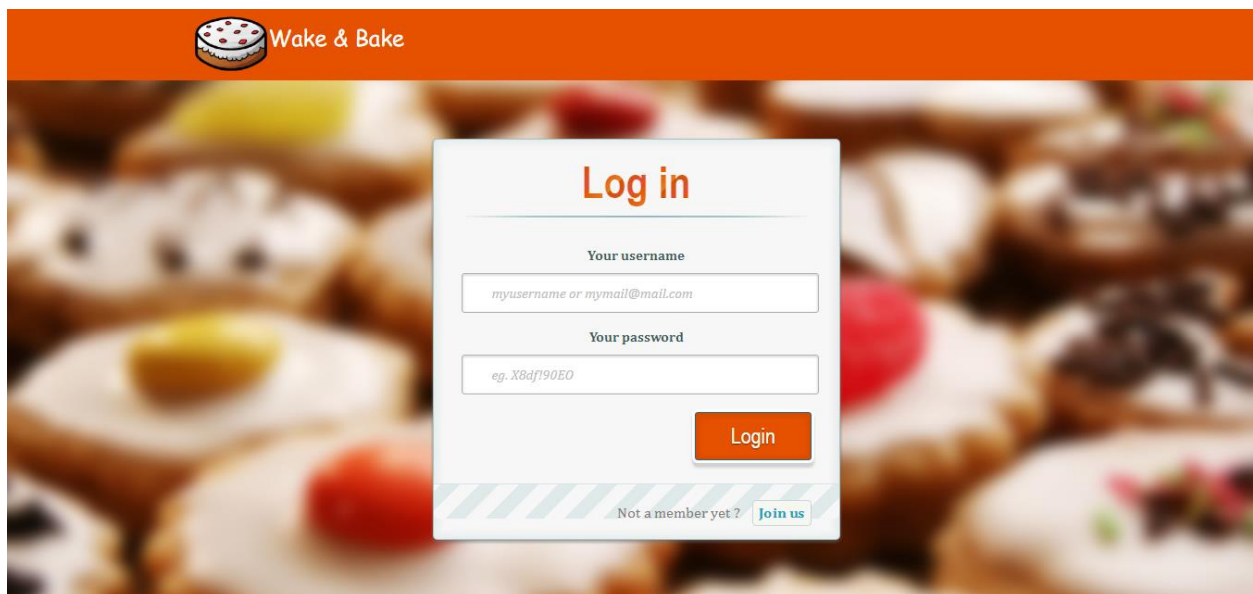
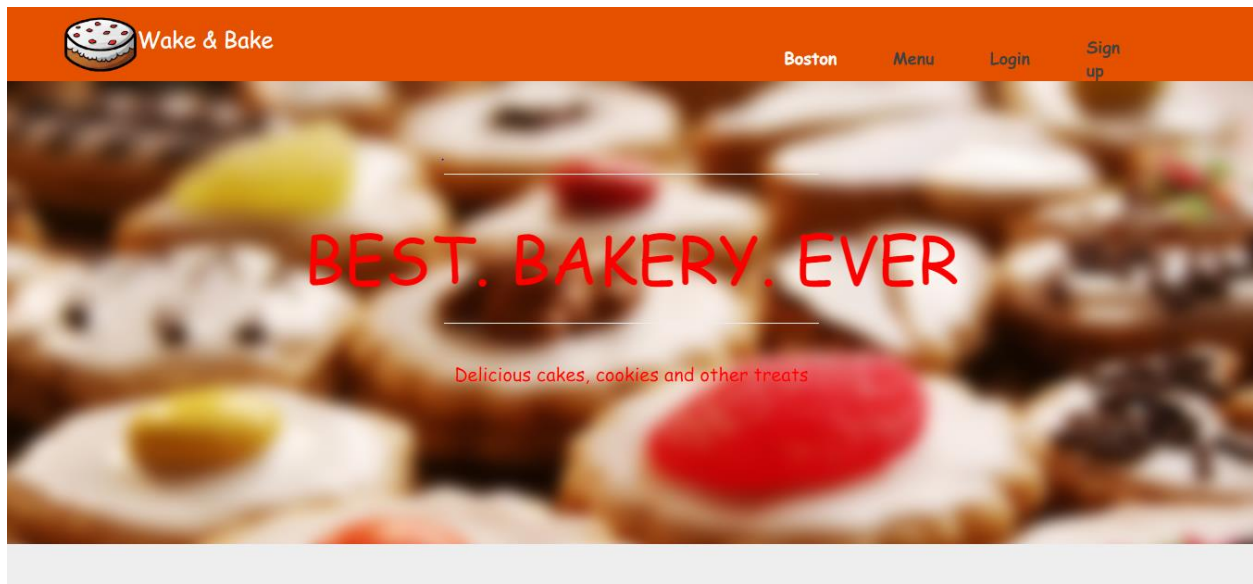
## Wake n' Bake


- Maven
- Network and Content Delivery
  - Route 53
  - VPC
- Compute
  - Elastic Beanstalk
  - EC2
- Server
  - Apache Tomcat 7
- Cloud Watch
- Security, Identity and Compliance
  - Certificate Manager
  - IAM
- Messaging
  - Simple Queue Service

## Web Application

---

- We have developed an online bakery system using Spring MVC framework and the link is <https://www.wakenbake.cc>
- We are using MySQL with RDS as a database and Aurora as the replica database.
- The application has a login mechanism with four user roles along with a payment page.
- We have developed the website which has four user roles as follows:
  - Admin
  - Chef
  - Customer
  - Delivery person



 Wake & Bake

## Registration Form

### Sign up

First Name

John

Last Name

Doe

Your email

mysupermail@mail.com

Age

25

Wake & Bake

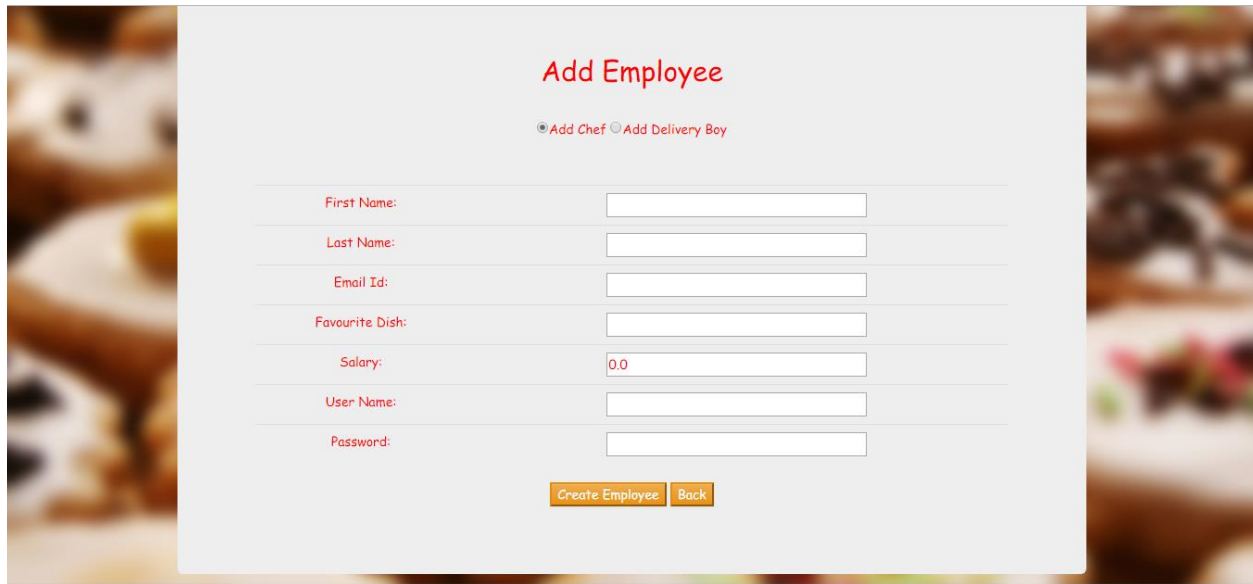
Logout

## Welcome Admin

Add Employee

Add Food Item to the Menu



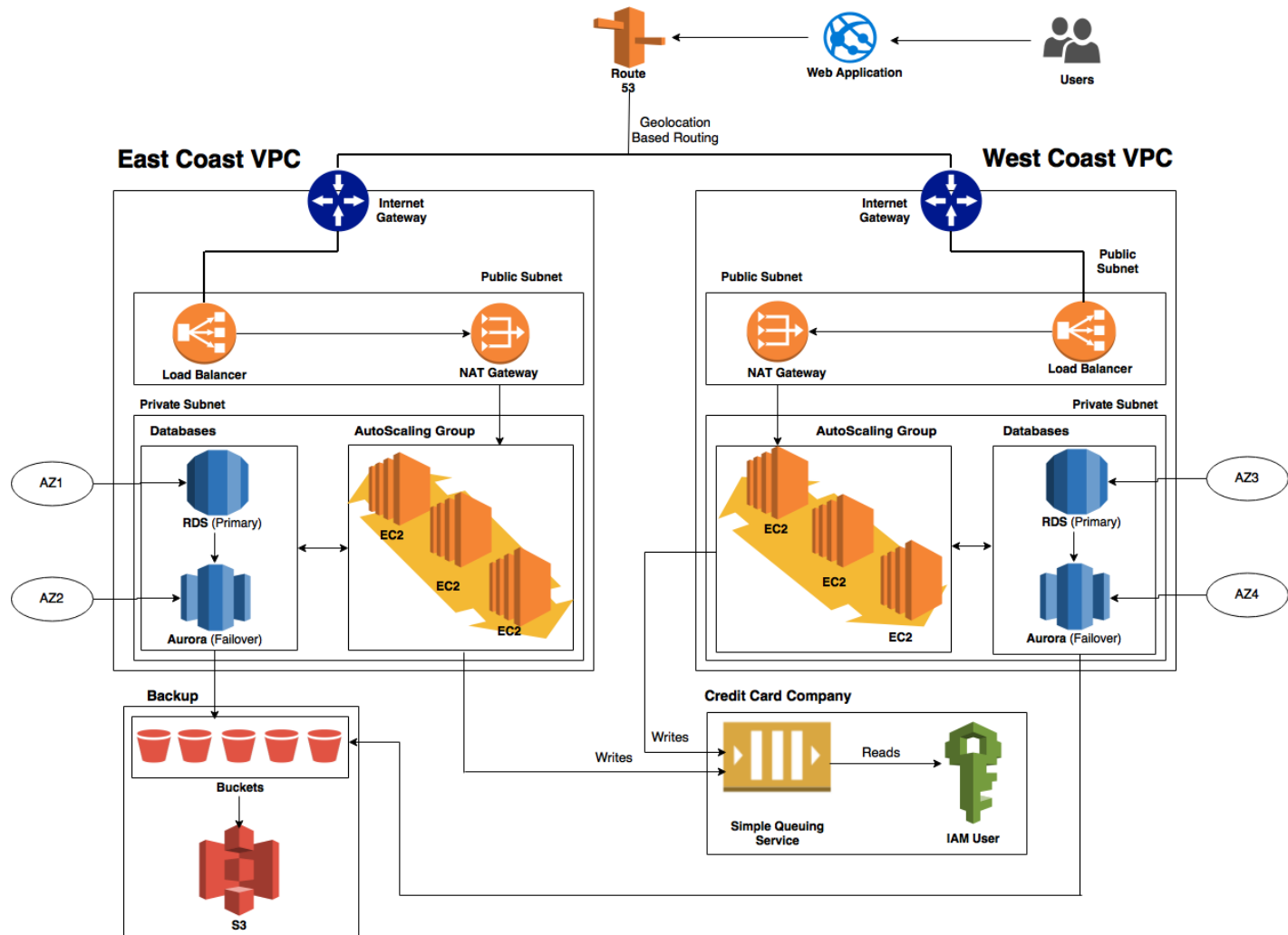
The image shows a web form titled "Add Employee" in red text. Below the title are two radio buttons: "Add Chef" (selected) and "Add Delivery Boy". The form contains seven input fields: "First Name:", "Last Name:", "Email Id:", "Favourite Dish:", "Salary:" (with "0.0" pre-filled), "User Name:", and "Password:". At the bottom are two orange buttons: "Create Employee" and "Back". The form is set against a light gray background with a blurred image of food on the sides.

**Add Employee**

☒ Add Chef ☐ Add Delivery Boy

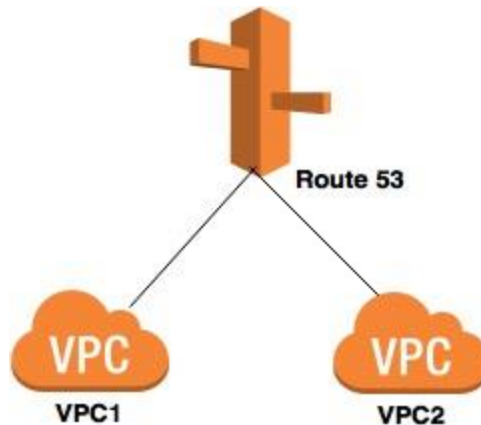
First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Email Id:	<input type="text"/>
Favourite Dish:	<input type="text"/>
Salary:	<input type="text" value="0.0"/>
User Name:	<input type="text"/>
Password:	<input type="text"/>

## CLOUD ARCHITECTURE



## Route 53

- Route 53 is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating domain names to IP addresses. To achieve low latency, we are using route 53's geolocation policy.



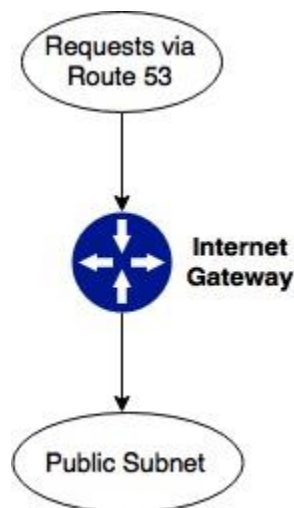
The screenshot shows the AWS Route 53 console. The left sidebar contains navigation links: Dashboard, Hosted zones (selected), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, and Pending requests. The main content area has a top bar with buttons: 'Back to Hosted Zones', 'Create Record Set' (highlighted in blue), 'Import Zone File', 'Delete Record Set', and 'Test Record Set'. Below this is a search bar for 'Record Set Name' and filters for 'Any Type' and 'Aliases Only'. A table titled 'Weighted Only' displays a list of record sets. The table has columns for 'type', 'Value', 'Evaluate Target Health', and 'Health'. The last record set is selected and highlighted in blue. To the right of the table is the 'Edit Record Set' panel. It contains fields for 'Name' (www.wakenbake.cc), 'Type' (CNAME - Canonical name), 'Alias' (Yes/No), 'TTL (Seconds)' (300), and 'Value' (awseb-e-v-AWSEBLoa-1T5890NR7C6U2-1193723084.us-east-1.elb.amazonaws.com). The 'Routing Policy' is set to 'Geolocation'. A 'Save Record Set' button is at the bottom right.

type	Value	Evaluate Target Health	Health
NS	ns-1320.awsdns-37.org. ns-189.awsdns-23.com. ns-2004.awsdns-58.co.uk. ns-572.awsdns-07.net.	-	-
OA	ns-1320.awsdns-37.org. awsdns-hostmaster.amazon	-	-
NAME	ec2-34-204-122-254.compute-1.amazonaws.com	-	-
NAME	awseb-e-v-AWSEBLoa-1T5890NR7C6U2-11937230	-	-

## Internet Gateway

---

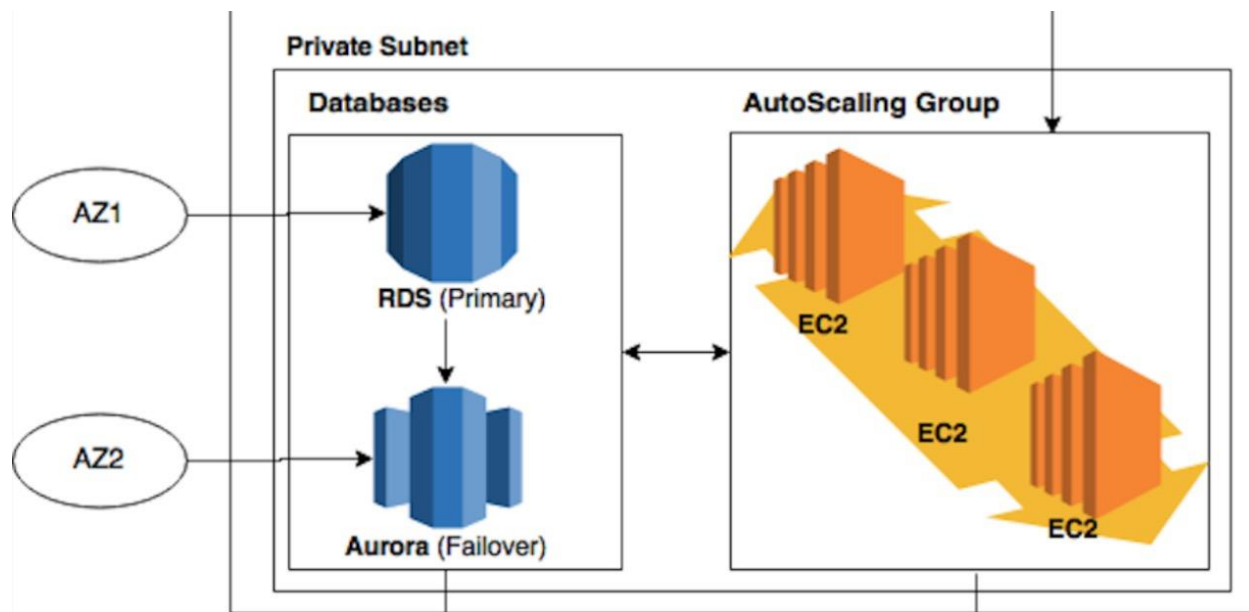
- An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet.
- We have setup two different internet gateways - one for each VPC to allow the communication of users with our application.
- Having an internet gateway does not affect availability or bandwidth need arising due to traffic.
- We are using Internet Gateway to allow inbound traffic to communicate to the public resources — Load Balancer in this case. 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.



## Elastic Load Balancer

---

- Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. This is helping us to achieve fault tolerance on EC2 instances on which our applications are running.
- To provide higher security to our application, we have deployed our elastic beanstalks with SSL certificates on HTTPS listeners for encryption ensuring that no request arriving at our applications are unauthorized to do so.
- ELB provides following features:
  - Detection of unhealthy EC2 instances.
  - Centralized management of SSL certificates.
  - Providing high scalability
  - Allowing scalability across different availability zones



Dashboard

Configuration

Logs

Health

Monitoring

Alarms

Managed Updates

Events

Tags

### Load Balancer

The following settings let you control the behavior of your environment's load balancer. [Learn more.](#)


Listener port: 80 ▾ The external facing HTTP port number to the load balancer.

Protocol: HTTP ▾ The protocol used by the listener.

Secure listener port: OFF ▾ The external facing HTTPS port number to the load balancer.

Protocol: HTTPS ▾ The protocol used by the secure listener.

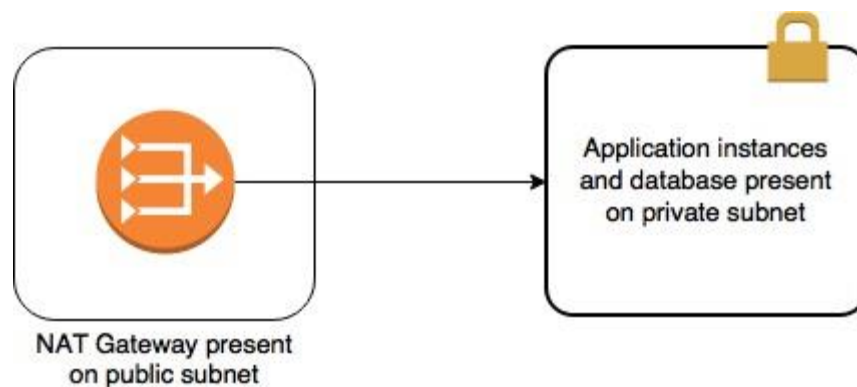
Cross-zone load balancing: ☐ Enable load balancing across multiple Availability Zones.

SSL certificate ID: (Optional) Select a certificate ▾ Refresh   
The Amazon Resource Name of the certificate to use for SSL.

### Connection Draining

## NAT Gateway

- NAT gateway ensures that the instances in private subnet can communicate with the internet or other internal AWS components while blocking the instantiation of communication with that instance.
- We have setup our NAT gateway on public subnet. This allows the communication of Load Balancer present on public subnet to communicate with EC2 instances present on private subnet.



## 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.
- Auto Scaling helps to maintain high availability of EC2 instances and allows to scale up or scale down the count of available resources based on network traffic.
- Incorporating this in our EC2 instance ensures that the users never face denial-of-service.
- Auto scaling also ensures that the desired number of instances are always up and running on the cloud.

### ▼ Auto Scaling

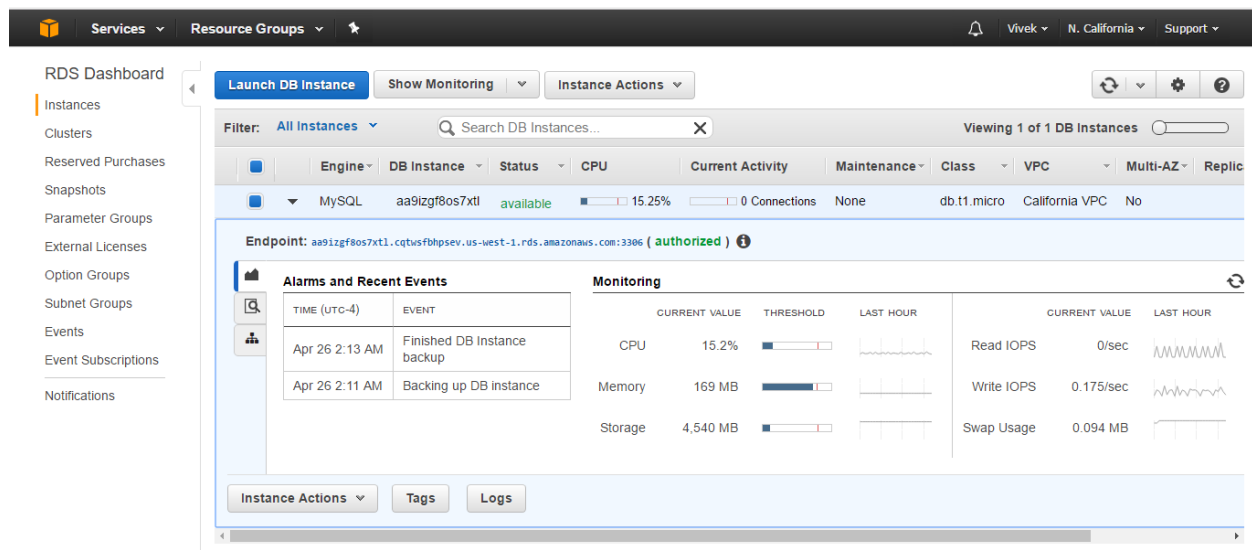
---

Use the following settings to control auto scaling behavior. [Learn more.](#)

Minimum instance count:	<input type="text" value="1"/>	Minimum number of instances to run.
Maximum instance count:	<input type="text" value="3"/>	Maximum number of instances to run.
Availability Zones:	<input type="text" value="Any 2"/>	Number of Availability Zones to run in.
Custom Availability Zones:	<input type="text" value="us-east-1a&lt;br/&gt;us-east-1b&lt;br/&gt;us-east-1c&lt;br/&gt;us-east-1d"/>	Specific Availability Zones to launch instances in.
Scaling cooldown (seconds):	<input type="text" value="10"/>	The amount of time after a scaling activity before any further trigger-related scaling activities can occur.

## Database

- We are employing AWS Relational Database as our primary database. This allows our POJOs to directly communicate with database server.
- To maintain the replica of our database, we have AWS Aurora Database present in different availability zone from RDS database. This database is the secondary database and will be useful in the case of failover.
- We are also taking snapshots of our databases at different point of time and storing it in buckets using the S3 feature of the AWS. This serves as the data dump for our entire architecture.





## Wake n' Bake

Amazon S3

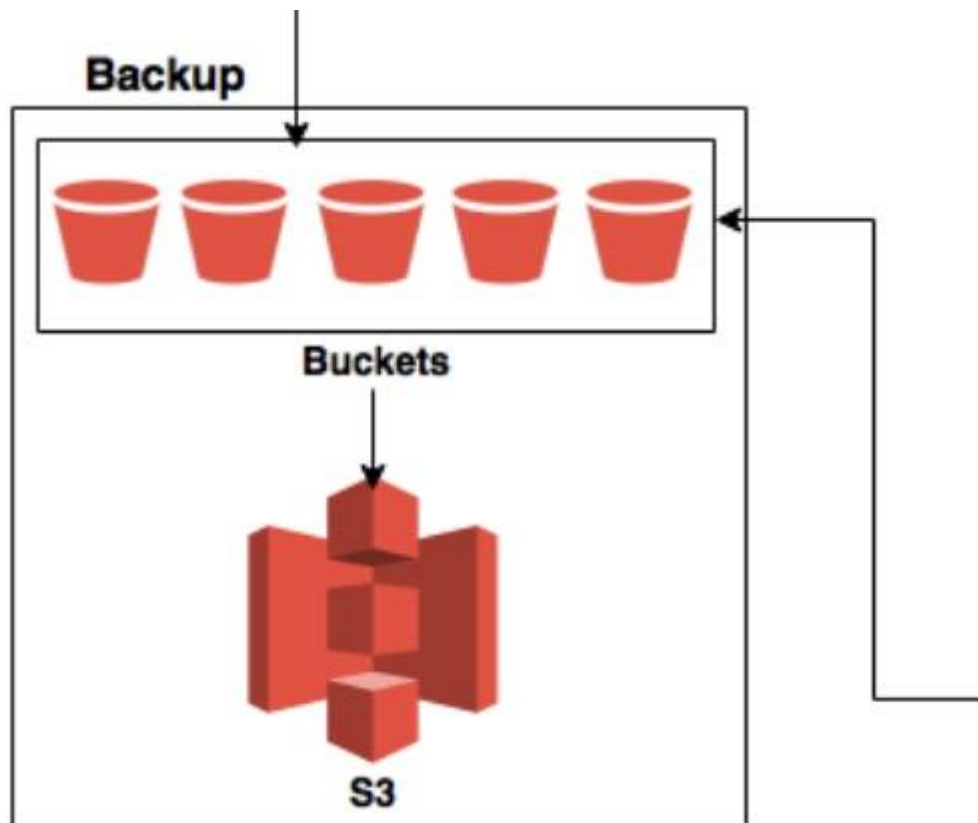
[Switch to the old console](#) [Discover the new console](#) [Quick tips](#)

Search for buckets

[+ Create bucket](#) [Delete bucket](#) [Empty bucket](#)

3 Buckets 3 Regions

Bucket name	Region	Date created
elasticbeanstalk-us-east-1-686578843959	US East (N. Virginia)	Feb 27, 2017 7:54:50 PM
elasticbeanstalk-us-west-1-686578843959	US West (N. California)	Mar 31, 2017 6:13:15 PM
rds-backup-waknbake	US West (Oregon)	Apr 16, 2017 8:29:56 PM

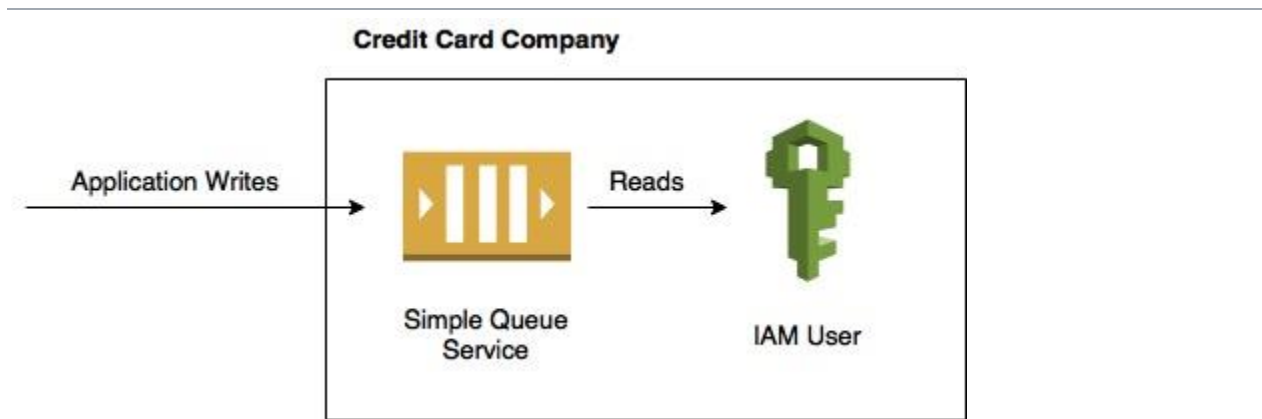


## Wake n' Bake

Objects				
Properties				
Permissions				
Management				
Q Type a prefix and press Enter to search. Press ESC to clear.				
Upload + Create folder More All Deleted objects US West (Oregon) ↺				
Viewing 1 to 100 >				
<input type="checkbox"/>	Name ↑	Last modified ↑	Size ↑	Storage class ↑
<input type="checkbox"/>	logBackup2017-04-17-01-33-35-F50D62EA2EE3C415	Apr 16, 2017 9:33:36 PM	361.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-01-40-41-E7FF6E731D2379D2	Apr 16, 2017 9:40:42 PM	334.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-02-26-03-CA5FCC562EDF6D4B	Apr 16, 2017 10:26:04 PM	394.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-02-40-44-510379279E69035F	Apr 16, 2017 10:40:45 PM	394.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-03-30-16-43A3778C15ABBD75	Apr 16, 2017 11:30:17 PM	395.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-03-33-47-1CC4ADB10F52DD68	Apr 16, 2017 11:33:48 PM	396.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-04-42-57-F55988447F0340B2	Apr 17, 2017 12:42:58 AM	394.0 B	Standard
<input type="checkbox"/>	logBackup2017-04-17-04-49-14-F92478CD8AD0E42B	Apr 17, 2017 12:49:15 AM	398.0 B	Standard

## Single queue system

- Fully-managed message queuing service for reliably communicating among distributed software components and micro-services
- Simple and cost-effective to decouple and coordinate the components of a cloud application
- Improves scalability and reliability, and is best practice design for modern applications
- Created an IAM user - Credit Card Company
  - Used by third party credit card employee to view and process transaction
  - Access restricted to only SQS



## CLOUD PRICING MODEL FOR THE NEXT 3-6 MONTHS

The estimated cost of Bake n' Wake Application is \$150.36 per month per zone using the AWS Simple Monthly Calculator. So, for 6 months the cost will be \$904.32.

The cost will also include and \$12 to purchase a domain first year.

So, the overall cost for our application, for 6 months, would come around \$916.32.

**amazon web services SIMPLE MONTHLY CALCULATOR** Language: English

Need Help? [Watch the Videos](#) or [Read How AWS Pricing Works](#)

Get Started with AWS: [Learn more about our Free Tier](#) or [Sign Up for an AWS Account](#)

**FREE USAGE TIER:** New Customers get free usage tier for first 12 months

Services	Estimate of your Monthly Bill (\$ 75.36)
<p><b>Estimate of Your Monthly Bill</b></p> <p>Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.</p> <p style="text-align: right;"><a href="#">Save and Share</a></p>	
Amazon S3	Amazon S3 Service (US-East) \$ 0.03
Amazon Route 53	Amazon Route 53 Service \$ 1.00
Amazon RDS	Amazon RDS Service (US-East) \$ 38.24
Amazon DynamoDB	Amazon VPC Service (US-East) \$ 36.60
Amazon ElastiCache	AWS Data Transfer In \$ 0.00
Amazon CloudWatch	AWS Data Transfer Out \$ 0.00
Amazon SES	AWS Support (Basic) \$ 0.00
Amazon SNS	<b>Free Tier Discount:</b> \$ -0.51
<b>Total Monthly Payment:</b>	<b>\$ 75.36</b>

**Common Customer Samples**

- Free Website on AWS
- AWS Elastic Beanstalk Default
- Marketing Web Site
- Large Web Application (All On-Demand)
- Media Application
- European Web Application
- Disaster Recovery and Backup

## KEY POINTS

---

- Safety
  - Independent database per zone
  - SQS
  - Aurora replica per RDS
- Security
  - Spring security
  - SSL certificate
  - Virtual Private Cloud
- Cost
  - Used AWS free tier components
  - Estimated monthly cost of around \$75.36 per zone
- Performance
  - Route 53 Geo-location (Latency)
  - Auto scaling over a specific CPU utilization
  - Handles more than 10k hits per minute

## CHALLENGES:

---

- Geo location traffic routing using Route 53
- Setting up VPC for both the zones (East and West)
- Handling security issues
- Designing a well-designed secured architecture
- Fail over traffic routing using Route 53

## REFERENCES

---

- [1] [www.whatisccloud.com](http://www.whatisccloud.com)
- [2] <https://aws.amazon.com>
- [3] [https://en.wikipedia.org/wiki/AWS\\_Elastic\\_Beanstalk](https://en.wikipedia.org/wiki/AWS_Elastic_Beanstalk)
- [4] [https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing)
- [5] [https://en.wikipedia.org/wiki/AWS\\_Elastic\\_Beanstalk](https://en.wikipedia.org/wiki/AWS_Elastic_Beanstalk)