

# Cloud Computing Project

Priyanka Bandekar  
Maansi Chandira  
Geetesh Pandit  
Siddharth Wale

## Requirements

1. Designing a high parallel log reporting which will consist of one main feature in which an increasing load on the application and decreases.
2. The entire application will be deployed in AWS CloudFront or Lambda function.
3. The entire interface will be displayed in AWS CloudFront.
4. Proper infrastructure and architecture is required to allow for a scaling of resources. This includes proper scaling of EC2 instances, Lambda functions, Amazon S3 storage and usage with user requirements.
5. User input fields are required.
6. There are some core features that need to be implemented which are:
  - a. Simulation of a user login to application and use 2 report using both 10 days.
  - b. Simulation of 10 user logins to application and use 2 report going back 30 days.
  - c. Generation of 10 user logins application and non 10 report going back 30 days.
  - d. Generation of 10 user logins application and non 10 report going back 60 days.
  - e. User input fields to allow me the ability to enter parameters for how many user logins I want and how many reports those users have and any other filtering.

## Auto Scaling

In our project we have used Auto Scaling which handles the auto scaling of EC2 instances. It is able to respond to the EC2 and Lambda implementation. When auto scaling done, instances can be terminated.

## CloudWatch

In our project we have used Amazon CloudWatch to monitor AWS resources such as Amazon EC2 instances and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate.

## Project Architecture



## Attached SSL Certificate to Load Balancer

The entire SSL certificate on the load balancer is present. All the settings to reflect off the HTTPS port (443) are set.



## Business Justification

We are in the business of renting cars to help people go wherever they want.

Demand for our cars varies largely according to various factors such as Day of week, Long Weekends, and sometimes even on Holiday Hours of day.

To satisfy our customer needs it is crucial for us to have a scalable architecture which could scale up and scale down as per business needs.

AWS provides a simple and cost efficient way to host our infrastructure on cloud.



## Cost Pricing

Cost Pricing model for the next 30 months (Converge Performance)

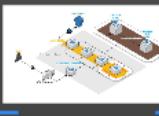


## Web Application

We have created an Car Rental Web Application. Wherein Customers can register and login on account. After choosing the car type and location they can book the rental and receive quote for the same.

All report page is holding data about the Cars from the database. Besides the main report we have a mechanism to print same.

## Visio Diagram



## Disaster Recovery

- Our infrastructure has been built on Amazon RDS which allows for having a master and redundant read replicas which is on standby.
- We typically have some pre-configured servers bundled as Amazon Machine Images (AMIs)
- Use Elastic IP addresses, which can be pre-allocated and identified in the preparation phase for DR, and associate them with your instances.
- Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances.
- An Amazon EBS snapshot to ensure that we have any installation packages and configuration information available in AWS.

## User Simulation

We are using a python script to create a web client which will open a socket connection to the server and request a web page from the application.

When we use any of the buttons on the report page to simulate user requests to the application, the web application creates a process to execute the python script.

Depending on the value of user count being simulated, the script opens that many connections to the servers and tries to fetch data.



## Use Case Snapshot



# Cloud Computing Project

Priyanka Bandekar  
Maansi Chandira  
Geetesh Pandit  
Siddharth Wale

## Requirements

1. Develop a login portal using Java Spring which will consist of user-case buttons to simulate an increasing load on the application and database.
2. The database can either be a relational database or a NoSQL database.
3. The application needs to be deployed on AWS.
4. Proper re-hausurcement and design to allow for auto-scaling of resources to accommodate the additional load in application, network, data storage and compute.
5. Minimum of one load balancer is required.
6. There are 5 use-case buttons that need to be implemented which are:
  - a. Simulation of 3 user login to application and run 2 report going back 10 days
  - b. Simulation of 10 user login to application and run 6 report going back 30 days
  - c. Simulation of 17 user login to application and run 10 report going back 60 days
  - d. Simulation of 24 user login to application and run 14 report going back 90 days
  - e. User input fields to allow me the ability to enter parameters for how many users are simulated and how many reports those users are querying site running.

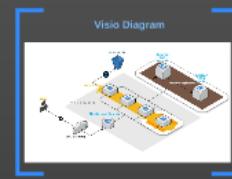
## Auto Scaling

In our project we have used AWS CloudWatch which handles the auto scaling feature. If any of the EC2 instances fail to respond, the CloudWatch detect it and launch a new instance. Other auto scale and cloud metrics can be terminated automatically.



## CloudWatch

In our project we have used Amazon CloudWatch to monitor AWS resources such as Amazon EC2 instances and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate.



## Business Justification

We are in the business of renting cars to help people go wherever they want.  
Demand for our cars varies largely according to different factors such as Day of week, Long Weekends and sometimes even on Happy Hours of day.

To satisfy our varying demands it is crucial for us to have a scalable architecture which could scale up and scale down as per business needs.

AWS provides a simple and cost efficient way to host our infrastructure on cloud.



## 6225-CSYE Network Structures and Cloud Computing

## Web Application

We have created a Car Rental Web Application where Customers can register and create an account. After creating an account, Customer can login to the portal and reserve data for him.

A report page is fetching data about the Cars from the database. Besides the main report, we have a mechanism to simulate users.

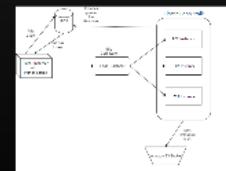
## User Simulation

We are using a python script to create a web client, which will open a socket connection to the server and request a web page from the application. When we use any of the buttons on the report page to simulate user requests to the application, the web application creates a process to execute the python script. Depending on the value of user count being simulated, the script opens that many connections to the server and tries to fetch data.

- Our infrastructure has been built allowing for having a master and redundancy. It is on standby.  
- We typically have some pre-configured Amazon Machine Images (AMIs).  
- Use Elastic IP addresses, which are identified in the preparation phase with them with your instances.  
- Use Elastic Load Balancing (ELB) with multiple instances.  
- An Amazon EBS snapshot to ensure installation packages and configurations in AWS.



## Project Architecture



6225-CSYE Network  
Structures and Cloud  
Computing

# Cloud Computing Project

Priyanka Bandekar  
Maansi Chandira  
Geetesh Pandit  
Siddharth Wale

# Requirements

1. Develop a login portal using Java Spring which will consist of use-case buttons to simulate an increasing load on the application and database.
2. The database can either be a relational database or a NoSQL database.
3. The entire stack must be deployed in AWS or Azure.
4. 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.
5. Minimum of one load balancer is required.
6. There are 5 use-case buttons that need to be implemented which are follows:
  - a. Simulation of 3 user login to application and run 2 report going back 10 days
  - b. Simulation of 10 user login to application and run 6 report going back 30 days
  - c. Simulation of 17 user login to application and run 10 report going back 60 days
  - d. Simulation of 24 user login to application and run 14 report going back 90 days.
  - e. User input fields to allow me the ability to enter parameters for how many users login simulation and how many reports those users are uniquely are running.

# Business Justification

We are in the business of renting cars to help people go wherever they want.

Demand for our cars varies largely according to various factors such as: Day of week, Long Weekends and sometimes even on Happy Hours of day .

To satisfy our varying demands it is crucial for us to have a scalable architecture which could scale up and scale down as per business needs.

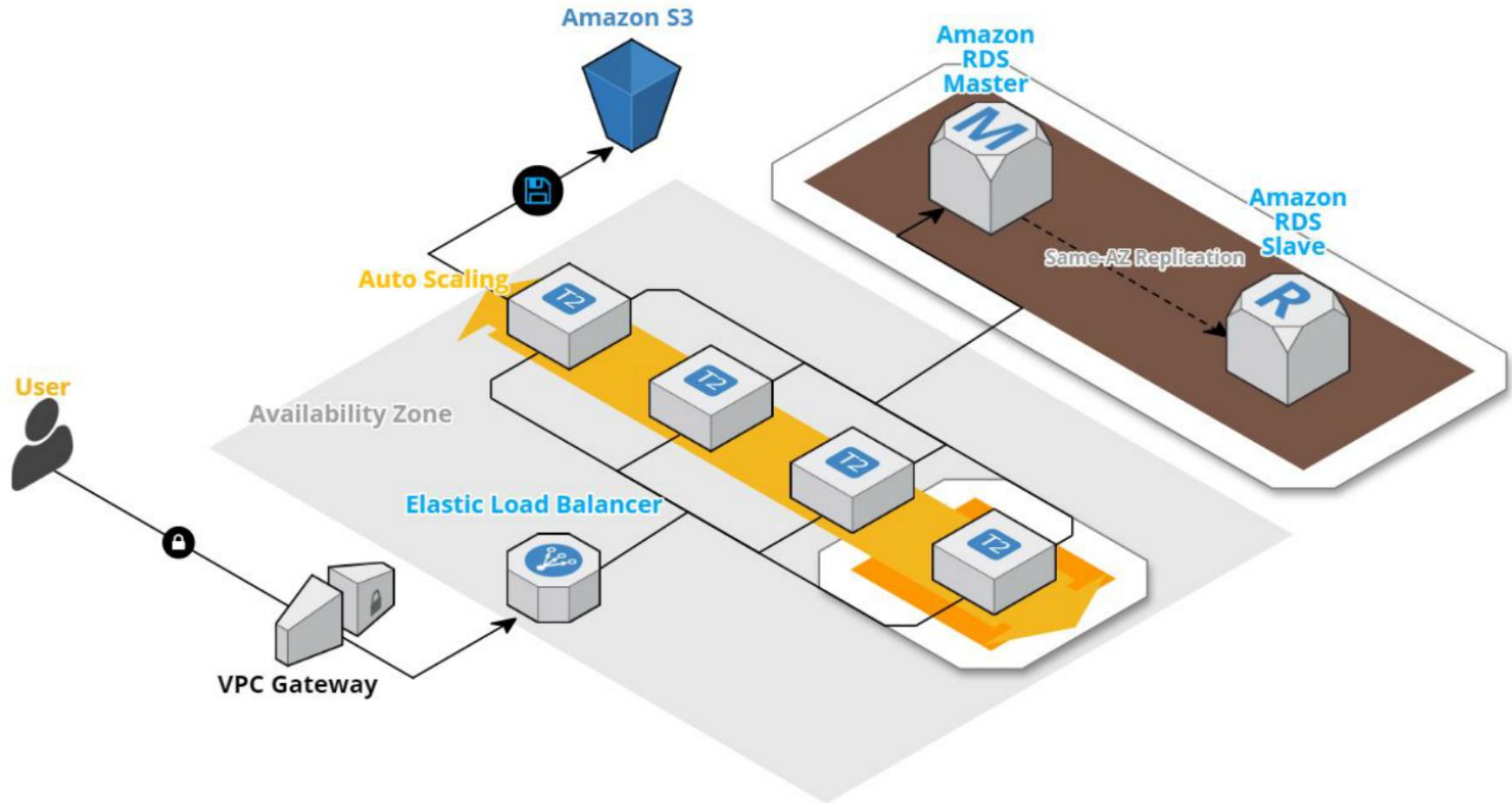
AWS provides a simple and cost efficient way to host our infrastructure on cloud.

# Web Application

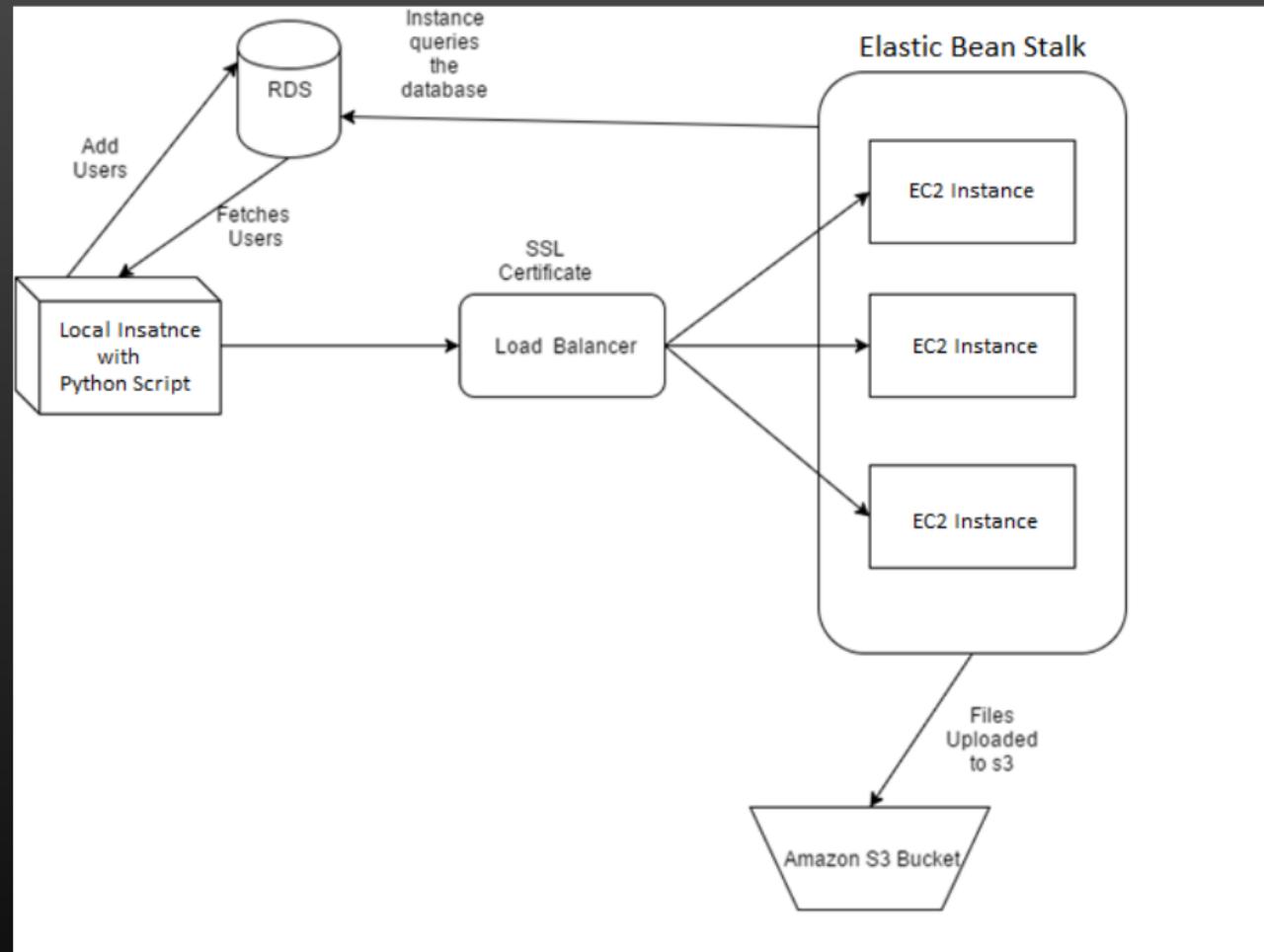
We have created a Car Rental Web Application where in Customers can register and create an account. After creating an account, Customer can login to the portal and reserve cars for rent.

A report page is fetching data about the Cars from the database. Besides the main report we have a mechanism to simulate users.

# Visio Diagram

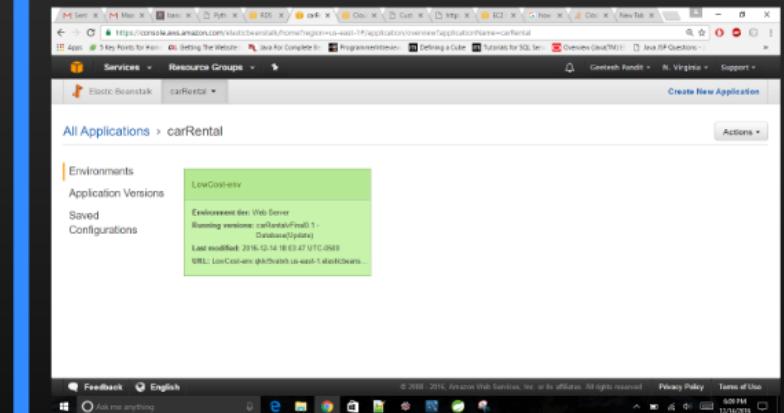
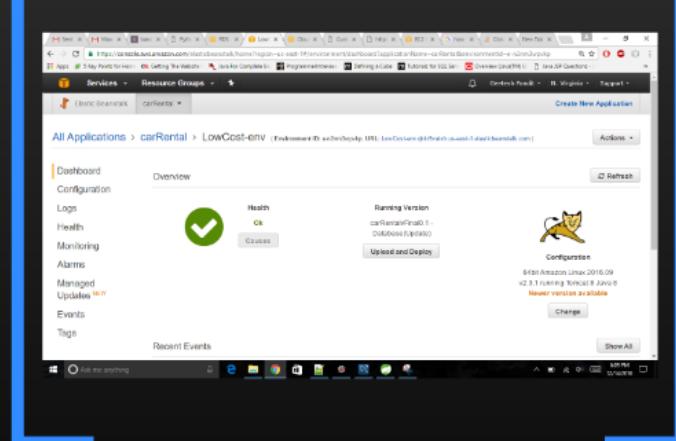


# Project Architecture



# Systems and Descriptions

*Launching instance using Elastic Beanstalk*



Sent X Mail X band X Python X RDS X LowC X CloudWatch Metrics X Cust X https X EC2 X how X Clos X New Tab X

https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=carRental&environmentId=e-n2nm3wpvxp

Apps 5 Key Points for Home CXL Getting The Website H Java For Complete Be ProgrammerIntreview M Defining a Cube m Tutorials for SQL Server Overview (Java(TM) EE Java JSP Questions - J

Services Resource Groups

Elastic Beanstalk carRental Create New Application

All Applications > carRental > LowCost-env (Environment ID: e-n2nm3wpvxp, URL: LowCost-env.qkkr9vatxh.us-east-1.elasticbeanstalk.com)

Actions Refresh

Dashboard Overview Refresh

Configuration

Logs

Health

Monitoring

Alarms

Managed Updates NEW

Events

Tags

Recent Events Show All

Health: Ok Causes

Running Version: carRentalvFinal0.1 - Database(Update)

Upload and Deploy

Configuration: 64bit Amazon Linux 2016.09 v2.3.1 running Tomcat 8 Java 8 Newer version available Change

Ask me anything 6:09 PM 12/14/2016

Sent X Maar X band X Python X RDS X carR X Cloud X Cust X https X EC2 X how X Clos X New Tab X

https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/application/overview?applicationName=carRental

Apps 5 Key Points for Home Getting The Website Java For Complete Be ProgrammerInterview Defining a Cube Tutorials for SQL Server Overview (Java(TM) EE Java JSP Questions - J

Services Resource Groups Create New Application

Elastic Beanstalk carRental Actions ▾

All Applications > carRental

Environments Application Versions Saved Configurations

LowCost-env

Environment tier: Web Server  
Running versions: carRentalvFinal0.1 - Database(Update)  
Last modified: 2016-12-14 18:03:47 UTC-0500  
URL: LowCost-env.qkkr9vatxh.us-east-1.elasticbeans...

Feedback English

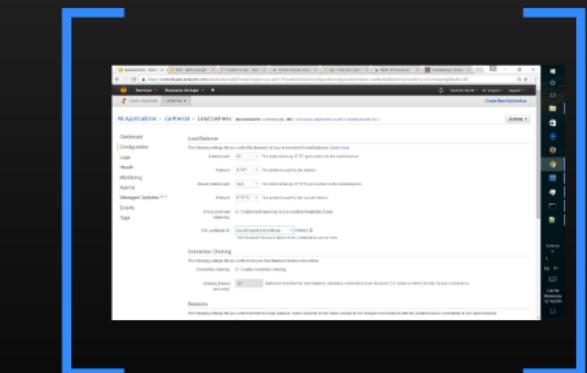
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy Terms of Use

6:09 PM 12/14/2016

## **Attached SSL Certificate to Load Balancer**

We added SSL certificate on the load balancer to override default EBS settings to redirect all the HTTP requests to HTTPS(443)



```

<VirtualHost *:80>
    <Proxy >
        <ProxyHTMLHandler allow
        Allow from all
    </Proxy>
    ProxyPass http://localhost:8080 encrypted
    ProxyPassReverse http://localhost:8080
    ProxyPassReverseCn on
    <Location />
        RewriteEngine On
        RewriteCond %{HTTP:X-Forwarded-Proto} https?
        RewriteRule /(.*) https://$1 [L,R]
    <Location /log>
        LogFormat "%b [%(Forwarded-For)] %u %t %v %V %D %B %Y %H %M %S" log
        ErrorLog /var/log/httpd/elasticsearch-matek-errord_log
        TransferLog /var/log/httpd/elasticsearch-matek-writes_log
        LogLevel warn
    </Location>

```

Screenshot of the AWS Elastic Beanstalk configuration page for the 'carRental' application environment.

The URL in the browser is: <https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/configuration?applicationName=carRental&environmentId=e-n2nm3wpvlp&edit=elb>

The left sidebar shows navigation links: Dashboard, Configuration (selected), Logs, Health, Monitoring, Alarms, Managed Updates (NEW), Events, and Tags.

The main content area is titled "Load Balancer". It includes the following settings:

- Listener port: 80 (The external facing HTTP port number to the load balancer)
- Protocol: HTTP (The protocol used by the listener)
- Secure listener port: 443 (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: cloudProjectFirstCertificate (The Amazon Resource Name of the certificate to use for SSL)

The "Connection Draining" section includes the following settings:

- Connection draining:  Enable connection draining
- Draining timeout (seconds): 20 (Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections)

The "Sessions" section includes the following setting:

- The following settings let you control whether the load balancer routes requests for the same session to the Amazon EC2 instance with the smallest load or consistently to the same instance.

The right side of the screen shows the Windows taskbar with various pinned icons and the system clock indicating 5:44 PM on Wednesday, 12/14/2016.

```
1 <VirtualHost *:80>
2   <Proxy *>
3     Order deny,allow
4     Allow from all
5   </Proxy>
6
7   ProxyPass / http://localhost:8080/ retry=0
8   ProxyPassReverse / http://localhost:8080/
9   ProxyPreserveHost on
10
11  RewriteEngine On
12  RewriteCond %{HTTP:X-Forwarded-Proto} !https
13  RewriteRule !/status https://{SERVER_NAME}{REQUEST_URI} [L,R]
14
15  LogFormat "%h (%{X-Forwarded-For}i) %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
16  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
17  TransferLog /var/log/httpd/elasticbeanstalk-access_log
18 </VirtualHost>
```

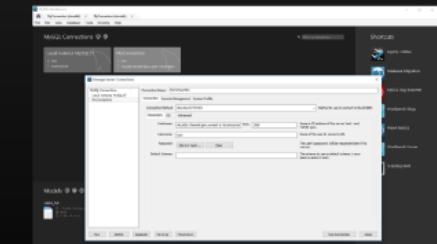
# *Amazon Relational Database Service (Amazon RDS)*

We have used RDS for MySQL Database Instance to fetch the data for our Application. Amazon RDS for MySQL gives access to the capabilities of a familiar MySQL database engine.

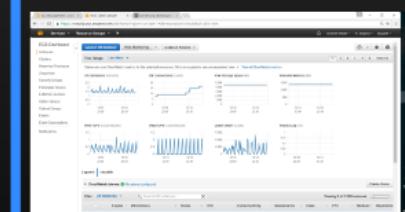
The RDS instance is created below along with the replica



We connected the RDS instance with MySQL database connection. We specified the end point URL of the DB instance with username and password that was used during launching of the DB instance.



RDS Utilization Graphs



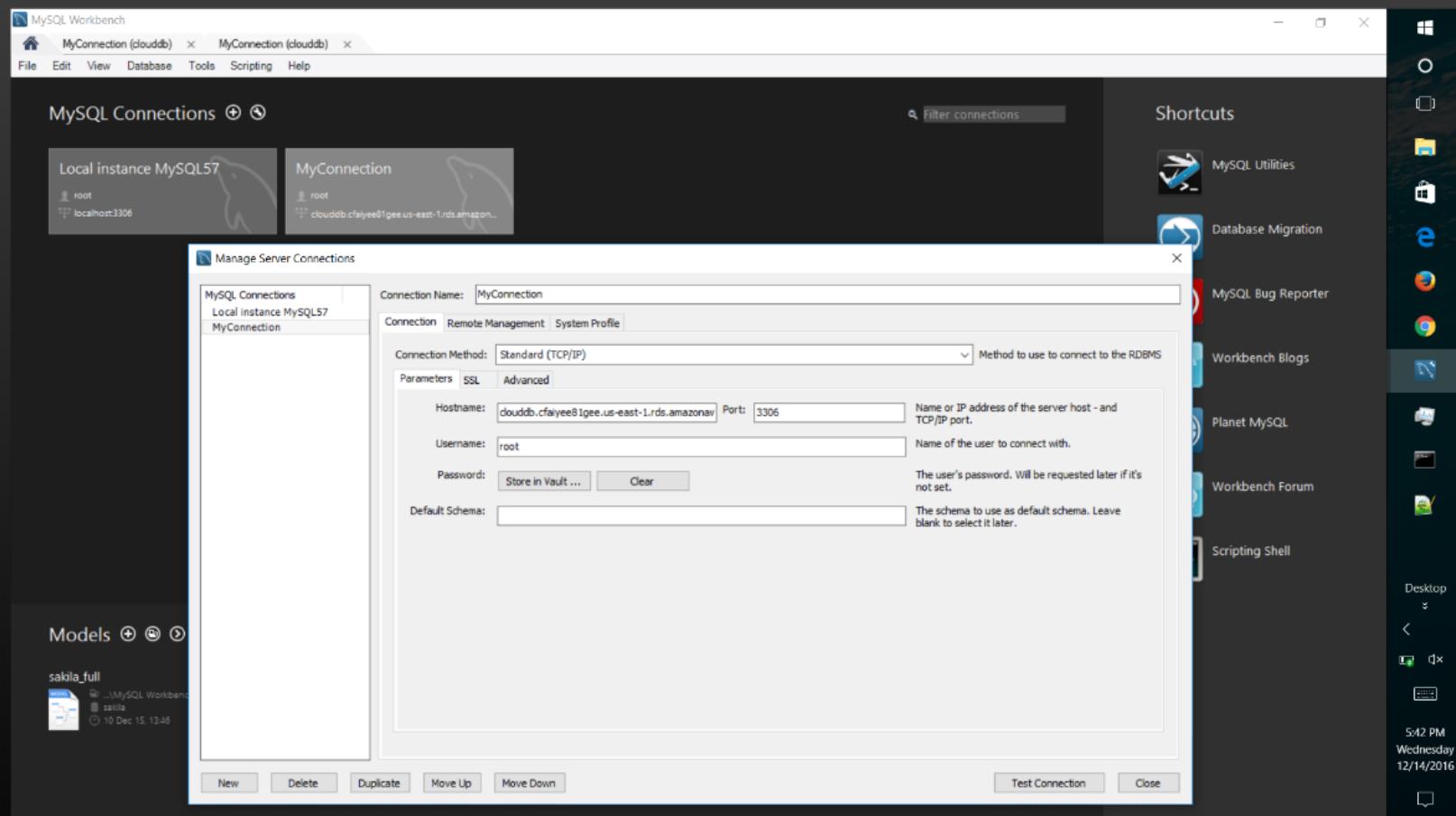
The RDS instance is created below along with the replica

The screenshot shows the AWS RDS Dashboard. On the left, there's a sidebar with options like Instances, Clusters, Reserved Purchases, Snapshots, Security Groups, Parameter Groups, External Licenses, Option Groups, Subnet Groups, Events, Event Subscriptions, and Notifications. The main area displays a table of DB instances. The table has columns for Engine, DB Instance, Status, CPU, Current Activity, Maintenance, Class, VPC, Multi-AZ, Replication Role, and Encrypted. There are two entries:

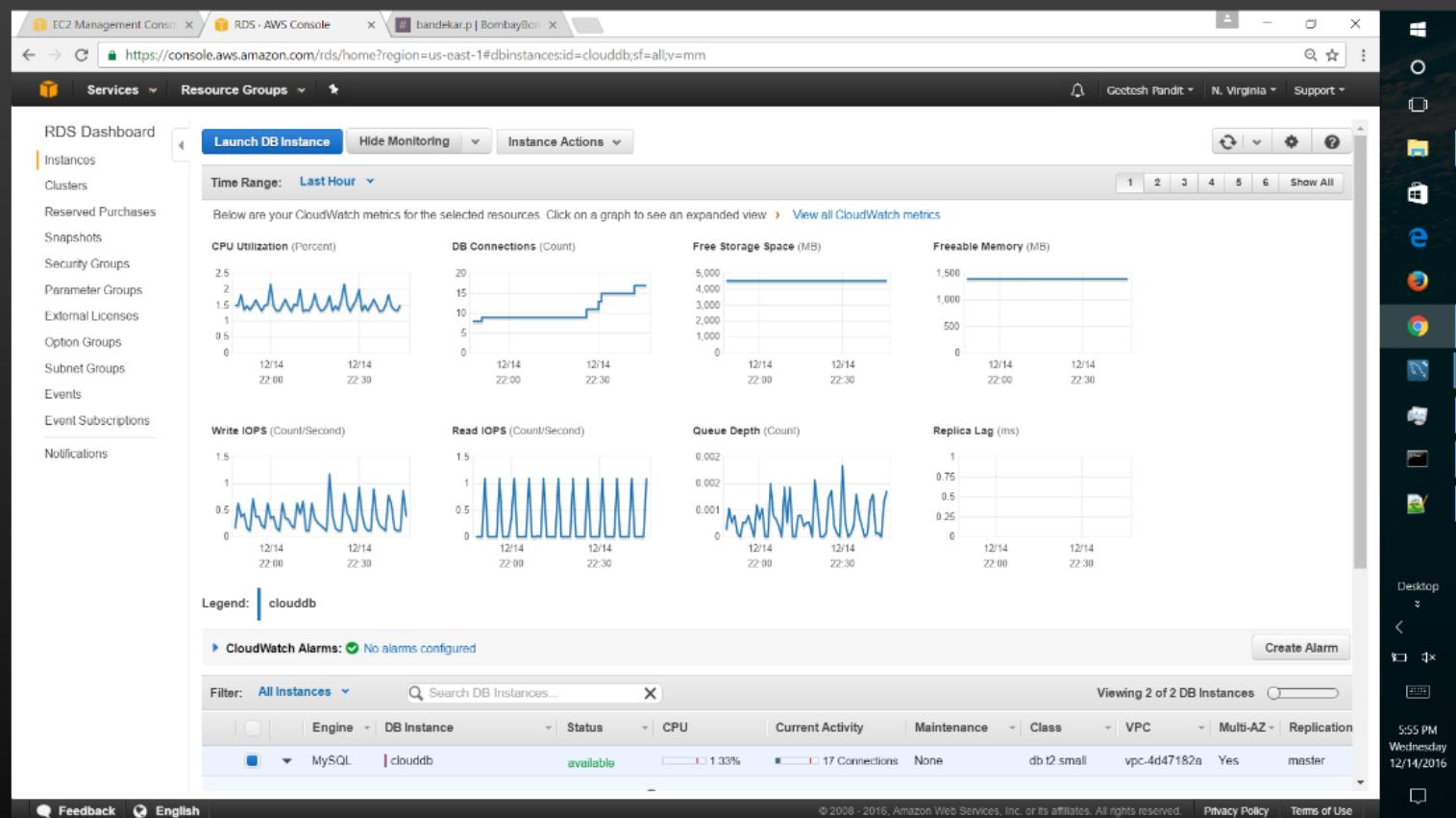
	Engine	DB Instance	Status	CPU	Current Activity	Maintenance	Class	VPC	Multi-AZ	Replication Role	Encrypted
1	MySQL	clouddb	available	1.33%	15 Connections	None	db.t2.small	vpc-4d47182a	Yes	master	No
2	MySQL	clouddbreplica	available	0.67%	0 Connections	None	db.t2.small	vpc-4d47182a	No	replica	No

At the bottom, there are links for Feedback, English, Privacy Policy, and Terms of Use. The status bar indicates the time as 5:34 PM on Wednesday, 12/14/2016.

**We connected the RDS instance with MySQL database connection. We specified the end point URL of the DB instance with username and password that was used during launching of the DB instance.**



# RDS Utilization Graphs



# Amazon Simple Storage Service (Amazon S3)

In Our Architecture we have created a S3 bucket wherein images are uploaded to it from our application.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs like 'Sent', 'Maar', '# maar', 'Python', 'RDS', 'S3 M', 'Cloud', 'Custom', 'https', 'EC2', 'Google', 'Close', 'Bootstrap', and 'X'. Below the navigation bar is a toolbar with 'Upload', 'Create Folder', 'Actions', a search bar ('Search by prefix'), and buttons for 'None', 'Properties', and 'Transfers'. The main area displays a table titled 'All Buckets / clouddb-images'. The table has columns for 'Name', 'Storage Class', 'Size', and 'Last Modified'. There are eight items listed:

Name	Storage Class	Size	Last Modified
1480542778605imgres.jpg	Standard	6.6 KB	Wed Dec 14 18:18:35 GMT-500 2016
14809687502381480542778605imgres.jpg	Standard	6.6 KB	Wed Dec 14 18:18:39 GMT-500 2016
14812184365502017-Mini-Cooper-S-Seven-Edition-PLACEMENT-626x382.jpg	Standard	44.5 KB	Wed Dec 14 18:18:36 GMT-500 2016
14812190419692017-Mini-Cooper-S-Seven-Edition-PLACEMENT-626x382.jpg	Standard	44.5 KB	Wed Dec 14 18:18:37 GMT-500 2016
14812192193512017-Mini-Cooper-S-Seven-Edition-PLACEMENT-626x382.jpg	Standard	44.5 KB	Wed Dec 14 18:18:38 GMT-500 2016
148167978417714809687502381480542778605imgres.jpg	Standard	135 bytes	Tue Dec 13 20:43:05 GMT-500 2016
1481683033466Application.PNG	Standard	135 bytes	Tue Dec 13 21:37:14 GMT-500 2016

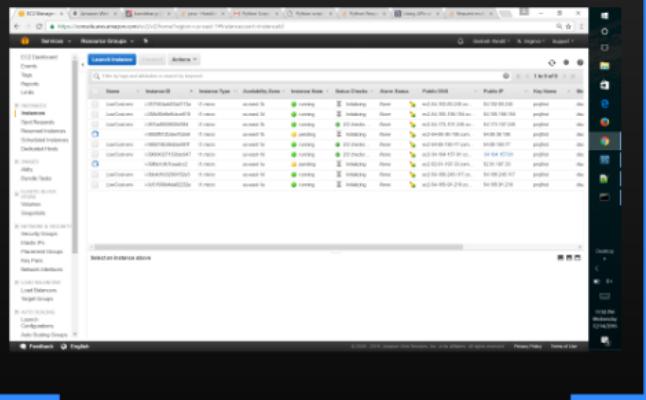
At the bottom of the page, there are links for 'Feedback', 'English', 'Privacy Policy', 'Terms of Use', and social media icons. The status bar at the very bottom shows '6:18 PM' and the date '12/14/2016'.

# Auto Scaling

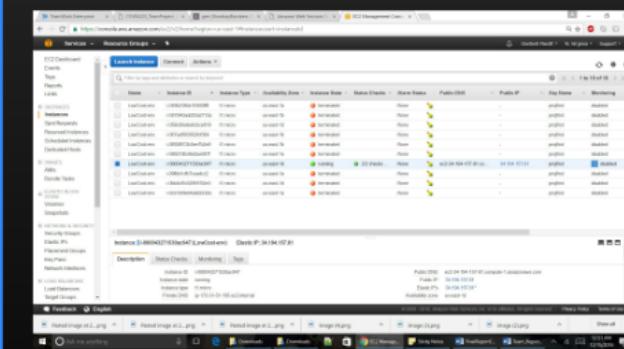
*In our project we have used AWS Elastic Beanstalk which handles the auto scaling feature itself.*

*If any of the EC2 instances fails to respond, the ELB will detect it and launch a replacement. When web traffic dies down, instances can be terminated automatically.*

*Scale Up*



*Scale Down*



# Scale Up

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation bar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays a table of running instances. The table columns are: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, Public IP, Key Name, and More. There are nine instances listed, all named 'LowCost-env' and all running in the 'us-east-1b' availability zone. The Public IP column shows various addresses like 54.162.60.245, 54.165.198.194, etc. The 'Actions' button at the top right of the table allows for instance management.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	More
LowCost-env	i-057540add33a2113a	t1.micro	us-east-1b	running	Initializing	None	ec2-54-162-60-245.co...	54.162.60.245	projfirst	disa
LowCost-env	i-058c90e8e8cbce619	t1.micro	us-east-1d	running	Initializing	None	ec2-54-165-198-194.co...	54.165.198.194	projfirst	disa
LowCost-env	i-067ad583562fcf304	t1.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-173-107-245.co...	54.173.107.245	projfirst	disa
	i-0692f612b3ee7b2e8	t1.micro	us-east-1b	pending	Initializing	None	ec2-54-89-36-106.com...	54.89.36.106	projfirst	disa
LowCost-env	i-08921b38dcbe087f	t1.micro	us-east-1a	running	2/2 checks ...	None	ec2-54-89-190-77.com...	54.89.190.77	projfirst	disa
LowCost-env	i-090043271530acf47	t1.micro	us-east-1d	running	2/2 checks ...	None	ec2-34-194-157-81.co...	34.194.157.81	projfirst	disa
	i-09fb1cfb7ceadcc2	t1.micro	us-east-1a	pending	Initializing	None	ec2-52-91-197-33.com...	52.91.197.33	projfirst	disa
LowCost-env	i-0bb4d1b32991f32e3	t1.micro	us-east-1d	running	Initializing	None	ec2-54-166-246-117.co...	54.166.246.117	projfirst	disa
LowCost-env	i-0c51506d4da82232e	t1.micro	us-east-1a	running	Initializing	None	ec2-54-165-91-219.co...	54.165.91.219	projfirst	disa

Select an instance above

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

1156 PM Wednesday 12/14/2016

# Scale Down

The screenshot shows the AWS EC2 Management Console interface. The left sidebar navigation bar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, and Load Balancing. The main content area displays a table of instances. One instance, "LowCost-env" (Instance ID: i-090043271530ac947), is currently running and highlighted with a blue selection bar. All other instances listed are in a terminated state. The bottom section provides detailed information for the selected instance, including its ID, state, type, and various network details.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
LowCost-env	i-049b238dc10450ff8	t1.micro	us-east-1a	terminated	None	None	-		projfirst	disabled
LowCost-env	i-057540add33a2113a	t1.micro	us-east-1b	terminated	None	None	-		projfirst	disabled
LowCost-env	i-058c90e8e8cbe619	t1.micro	us-east-1d	terminated	None	None	-		projfirst	disabled
LowCost-env	i-067ad503562fcf384	t1.micro	us-east-1b	terminated	None	None	-		projfirst	disabled
LowCost-env	i-0692f612b3ee7b2e8	t1.micro	us-east-1b	terminated	None	None	-		projfirst	disabled
LowCost-env	i-089218b38dcbe087f	t1.micro	us-east-1a	terminated	None	None	-		projfirst	disabled
LowCost-env	i-090043271530ac947	t1.micro	us-east-1d	running	2/2 checks ...	None	ec2-34-194-157-81.co...	34.194.157.81	projfirst	disabled
LowCost-env	i-09fb1cfb7ceadcc2	t1.micro	us-east-1a	terminated	None	None	-		projfirst	disabled
LowCost-env	i-0bb4d1b32991f32e3	t1.micro	us-east-1d	terminated	None	None	-		projfirst	disabled
LowCost-env	i-0c51506d4da82232e	t1.micro	us-east-1a	terminated	None	None	-		projfirst	disabled

Instance: i-090043271530ac947 (LowCost-env)    Elastic IP: 34.194.157.81

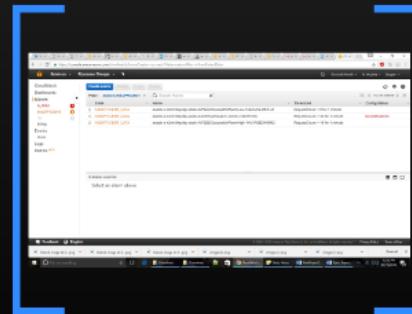
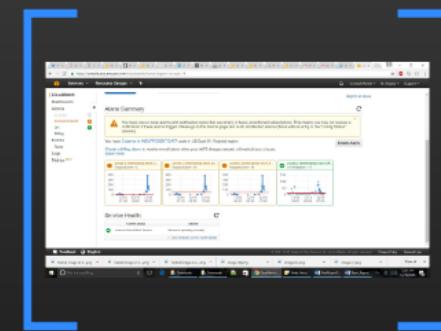
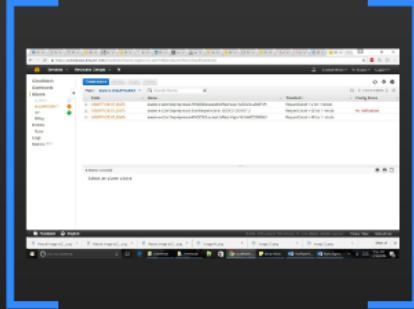
Description	Status Checks	Monitoring	Tags
Instance ID: i-090043271530ac947	Public DNS: ec2-34-194-157-81.compute-1.amazonaws.com	Public IP: 34.194.157.81	
Instance state: running	Elastic IPs: 34.194.157.81*	Availability zone: us-east-1d	
Instance type: t1.micro			
Private DNS: ip-172-31-51-165.ec2.internal			

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Pasted image at 2...png Pasted image at 2...png Pasted image at 2...png image (4).png image (3).png image (2).png Show all 12:51 AM 12/15/2016

# *CloudWatch*

In our project we have used Amazon CloudWatch to monitor AWS resources such as Amazon EC2 instances and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate.



https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarm:alarmFilter=inInsufficientData

Services ▾ Resource Groups ▾

CloudWatch Dashboards Alarms ALARM 1 INSUFFICIENT 3 OK 0 Billing Events Rules Logs Metrics NEW

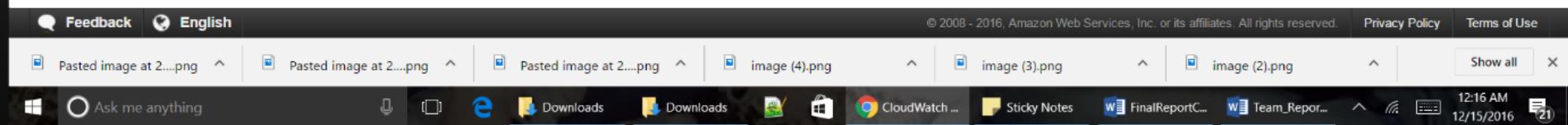
Create Alarm Modify Copy Delete

Filter: State is INSUFFICIENT Search Alarms

State	Name	Threshold	Config Status
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-AWSEBCloudwatchAlarmLow-1UEAUCLJMHFJA	RequestCount < 2 for 1 minute	
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-SumRequestAlarm-1JD3C31DGKGYU	RequestCount > 50 for 1 minute	No notifications
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-AWSEBCloudwatchAlarmHigh-1KUVA9ED9H9M2	RequestCount > 48 for 1 minute	

0 Alarms selected

Select an alarm above



https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarm:alarmFilter=inInsufficientData

Services Resource Groups

CloudWatch Dashboards Alarms ALARM INSUFFICIENT OK Billing Events Rules Logs Metrics NEW

Create Alarm Modify Copy Delete

Filter: State is INSUFFICIENT Search Alarms

State	Name	Threshold	Config Status
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-AWSEBCloudwatchAlarmLow-1UEAUCLJMHFJA	RequestCount < 2 for 1 minute	
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-SumRequestAlarm-1JD3C31DGKGYU	RequestCount > 50 for 1 minute	No notifications
INSUFFICIENT_DATA	awseb-e-n2nm3wpvkp-stack-AWSEBCloudwatchAlarmHigh-1KUVA9ED9H9M2	RequestCount > 48 for 1 minute	

0 Alarms selected Select an alarm above

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Pasted image at 2....png Pasted image at 2....png Pasted image at 2....png image (4).png image (3).png image (2).png Show all

Ask me anything Downloads Downloads CloudWatch... Sticky Notes FinalReportC... Team\_Report... 12:22 AM 12/15/2016 21

https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#

Services ▾ Resource Groups ▾

CloudWatch

Dashboards

Alarms

ALARM

INSUFFICIENT 3

OK 1

Billing

Events

Rules

Logs

Metrics NEW

Report an Issue

## Alarm Summary

You have one or more alarms with notification topics that are empty or have unconfirmed subscriptions. This means you may not receive a notification if these alarms trigger. Please go to the Alarms page and audit all affected alarms (those with an entry in the "Config Status" column).

You have 3 alarms in INSUFFICIENT DATA state in US East (N. Virginia) region.

Create a billing alarm to receive e-mail alerts when your AWS charges exceed a threshold you choose.

Learn more

Create Alarm

awseb-e-n2nm3wpvkp-stack-A... RequestCount < 2

awseb-e-n2nm3wpvkp-stack-Su... RequestCount > 50

awseb-e-n2nm3wpvkp-stack-A... RequestCount > 48

awseb-e-n2nm3wpvkp-stack-CP... CPUUtilization > 10

## Service Health

Current Status	Details
<span>✓</span> Amazon CloudWatch Service	Service is operating normally <a href="#">View complete service health details</a>

Feedback English

Pasted image at 2....png Pasted image at 2....png Pasted image at 2....png image (4).png image (3).png image (2).png Show all

Ask me anything 12:24 AM 12/15/2016

Downloads Downloads CloudWatch ... Sticky Notes FinalReportC... Team\_Report...

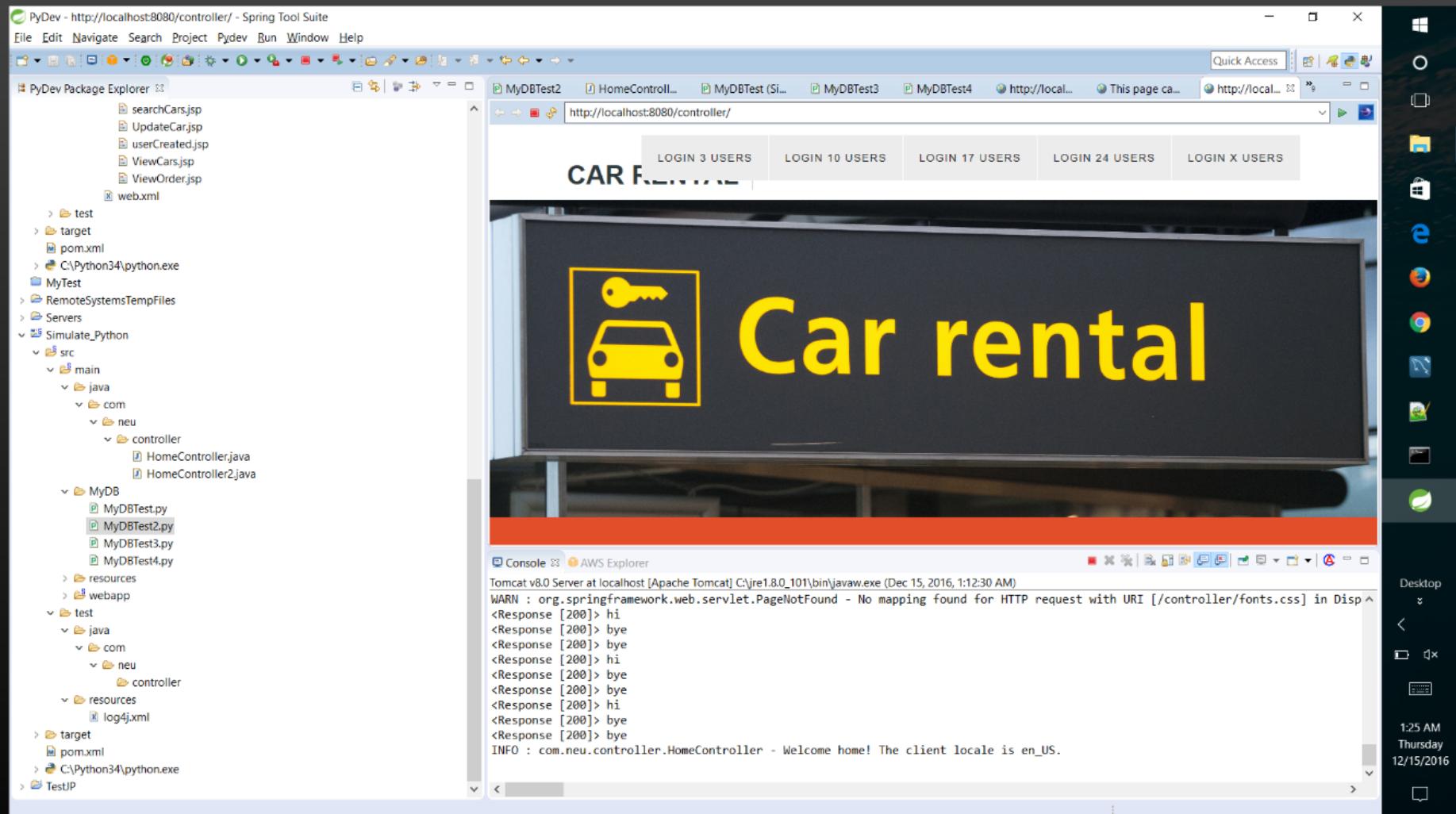
# User Simulation

We are using a python script to create a web client, which will open a socket connection to the server and request a web page from the application.

When we use any of the buttons on the report page to simulate user requests to the application, the web application creates a process to execute the python script.

Depending on the value of user count being simulated, the script opens that many connections to the server and tries to fetch data.

# Use Case Snapshot



# Cost Pricing model for the next 3-6 months (Cost vs Performance)

## AWS Simple Monthly Calculator

calculator.s3.amazonaws.com/index.html

Language: English

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

Reset All

Services Estimate of your Monthly Bill (\$ 65.15)

Choose region: US-East / US Standard (Virginia)

Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.

Compute: Amazon EC2 Instances:

Description	Instances	Usage	Type	Billing Option	Monthly Cost
LowCost-env	1	24 Hours/Day	Linux on t1.micro	On-Demand (No Co	\$ 14.64

Add New Row

Compute: Amazon EC2 Dedicated Hosts:

Description	Number of Hosts	Usage	Type	Billing Option

Add New Row

Storage: Amazon EBS Volumes:

Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput	Snapshot Storage
vol-06d239d	1	General Purpose SSD (gp2)	8 GB	100	128 MBs/sec	0 GB-month of Storage

Add New Row

Elastic IP:

Number of Additional Elastic IPs: 1

Elastic IP Non-attached Time: 0 Hours/Month

Number of Elastic IP Remaps: 0 Per Month

Data Transfer:

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

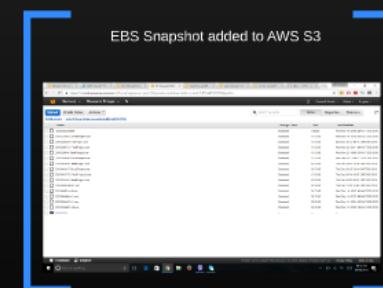
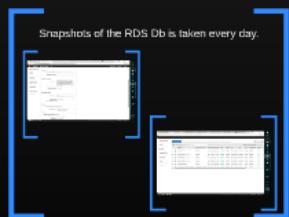
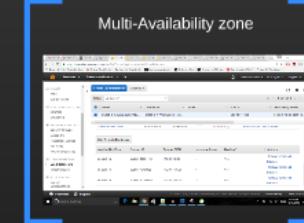
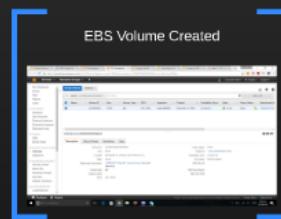
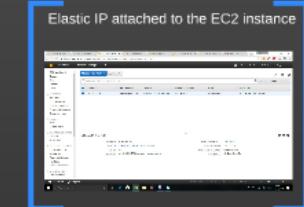
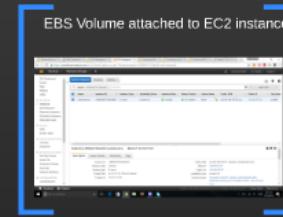
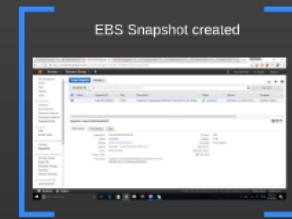
Pasted image at 2...png Pasted image at 2...png Pasted image at 2...png image (4).png image (3).png image (2).png Show all

Ask me anything e Downloads Downloads Amazon We... Sticky Notes FinalReportC... Team\_Report... 1:12 AM 12/15/2016

# Disaster Recovery

- Our infrastructure has been built on Amazon RDS which allows for having a master and redundant read replica which is on standby.
- We typically have some pre-configured servers bundled as Amazon Machine Images (AMIs)
- Use Elastic IP addresses, which can be pre-allocated and identified in the preparation phase for DR, and associate them with your instances.
- Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances.
- An Amazon EBS snapshot to ensure that we have any installation packages and configuration information available in AWS.

# Disaster Recovery



# EBS Snapshot created

The screenshot shows the AWS EC2 Management console interface. The left sidebar navigation bar includes links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, Bundle Tasks, Volumes, Snapshots (which is the selected category), Network & Security, Load Balancing, and Feedback. The main content area displays a table of snapshots under the heading "Create Snapshot". A single row is visible, showing a snapshot named "snap-0f33567f8e5549470" which is 8 GiB in size and was created by "CreateImage(i-090043271530ac947) for ami-28a9a...". The status is "completed" and it started on December 14, 2016 at 9:02:07 UTC-5, with a progress of 100% and an available (100%) capacity. The volume is "vol-06d239de4929a06e5" and it is not encrypted. The description field indicates it was created from the same ami-28a9a63f instance. Below the table, a detailed view for the selected snapshot is shown with tabs for Description, Permissions, and Tags. The "Description" tab is active, displaying the same information as the table.

Name	Snapshot ID	Size	Description	Status	Started	Progress
snap-0f33567f8e5549470	8 GiB	Created by CreateImage(i-090043271530ac947) for ami-28a9a...	completed	December 14, 2016 at 9:02:07 UTC-5	available (100%)	

**Snapshot: snap-0f33567f8e5549470**

**Description** **Permissions** **Tags**

Snapshot ID	snap-0f33567f8e5549470	Progress	100%
Status	completed	Capacity	8 GiB
Volume	vol-06d239de4929a06e5	Encrypted	Not Encrypted
Started	December 14, 2016 at 9:02:07 PM UTC-5	KMS Key ID	
Owner	497443303796	KMS Key Aliases	
Product codes	-	KMS Key ARN	
Description	Created by CreateImage(i-090043271530ac947) for ami-28a9a63f from vol-06d239de4929a06e5		

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

# EBS Volume Created

The screenshot shows the AWS EC2 Management console with the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volumes:search=vol-06d239de4929a06e5;sort=desc:createTime>. The left sidebar is collapsed, and the main area displays the 'Create Volume' tab of the Volumes section. A search bar at the top of the table lists 'vol-06d239de4929a06e5'. The table headers are Name, Volume ID, Size, Volume Type, IOPS, Snapshot, Created, Availability Zone, State, Alarm Status, and Attachment Info. One row is visible, showing a volume named 'vol-06d239de4929a06e5' with a size of 8 GiB, type gp2, and IOPS of 100 / 3000. It was created on December 14, 2016, in the us-east-1d availability zone, is in-use, and has no alarm status. The attachment information shows it is attached to instance 'i-090043271530ac947' with device '/dev/sda1'. Below the table, a detailed view of the volume 'vol-06d239de4929a06e5' is shown under the 'Description' tab. The volume details include its ID, size (8 GiB), creation date (December 14, 2016), state (in-use), and attachment information. The volume type is gp2, and it has an IOPS of 100 / 3000. The detailed view also shows the snapshot (snap-0a8fd85f9d27f1e09), availability zone (us-east-1d), and KMS key information.

Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment Info
vol-06d239de4929a06e5	8 GiB	gp2	100 / 3000	snap-0a8fd85f9d27f1e09	December 14, 2016	us-east-1d	in-use	None	i-090043271530ac947	

**Volumes: vol-06d239de4929a06e5**

**Description** **Status Checks** **Monitoring** **Tags**

Volume ID	vol-06d239de4929a06e5	Alarm status	<i>None</i>
Size	8 GiB	Snapshot	<a href="#">snap-0a8fd85f9d27f1e09</a>
Created	December 14, 2016 at 3:00:38 PM UTC-5	Availability Zone	us-east-1d
State	in-use	Encrypted	Not Encrypted
Attachment information	<a href="#">i-090043271530ac947 (LowCost-env) /dev/sda1 (attached)</a>	KMS Key ID	
Volume type	gp2	KMS Key Aliases	
Product codes	-	KMS Key ARN	
IOPS	100 / 3000		

# EBS Volume attached to EC2 instance

The screenshot shows the AWS EC2 Management console interface. The left sidebar navigation includes 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES' (selected), 'Instances', 'Spot Requests', 'Reserved Instances', 'Scheduled Instances', 'Dedicated Hosts', 'IMAGES', 'AMIs', 'Bundle Tasks', 'ELASTIC BLOCK STORE', 'Volumes', 'Snapshots', 'NETWORK & SECURITY', 'Security Groups', 'Elastic IPs', 'Placement Groups', 'Key Pairs', 'Network Interfaces', and 'LOAD BALANCING', 'Load Balancers'. The main content area displays the 'Launch Instance', 'Connect', and 'Actions' buttons. A search bar shows 'search : i-090043271530ac947'. The table lists one instance: 'LowCost-env' (Instance ID: i-090043271530ac947, Instance Type: t1.micro, Availability Zone: us-east-1d, Status: running, 2/2 checks passed, Alarm Status: None, Public DNS: ec2-34-194-157-81.compute-1.amazonaws.com, Public IP: 34.194.157.81, Key Name: projfirst). Below the table, the instance details for 'i-090043271530ac947 (LowCost-env)' are shown, including its configuration (Instance ID: i-090043271530ac947, Instance state: running, Instance type: t1.micro, Private DNS: ip-172-31-51-165.ec2.internal, Private IPs: 172.31.51.165) and network information (Public DNS: ec2-34-194-157-81.compute-1.amazonaws.com, Public IP: 34.194.157.81\*, Elastic IPs: 34.194.157.81\*, Availability zone: us-east-1d, Security groups: rds-launch-wizard-3, awseb-e-n2nm3wpvlp-stack-AWSEBSecurityGroup-1D74IEMPYGAVW, view inbound rules).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
LowCost-env	i-090043271530ac947	t1.micro	us-east-1d	running	2/2 checks ...	None	ec2-34-194-157-81.co...	34.194.157.81	projfirst

Instance: i-090043271530ac947 (LowCost-env)    Elastic IP: 34.194.157.81

Description	Status Checks	Monitoring	Tags
Instance ID: i-090043271530ac947	Public DNS: ec2-34-194-157-81.compute-1.amazonaws.com		
Instance state: running	Public IP: 34.194.157.81		
Instance type: t1.micro	Elastic IPs: 34.194.157.81*		
Private DNS: ip-172-31-51-165.ec2.internal	Availability zone: us-east-1d		
Private IPs: 172.31.51.165	Security groups: rds-launch-wizard-3, awseb-e-n2nm3wpvlp-stack-AWSEBSecurityGroup-1D74IEMPYGAVW, view inbound rules		

Feedback English Ask me anything © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use 10:57 PM 12/14/2016

# EBS Snapshot added to AWS S3

The screenshot shows the AWS S3 Management console interface. At the top, there are several tabs: Disaster Recovery, AWS Disaster Rec, EC2 Management, S3 Management (which is active), Copying an AMI, Launching an Inst, Create an AMI fro, and an inbox tab. The URL in the browser is https://console.aws.amazon.com/s3/home?region=us-east-1&bucket=elasticbeanstalk-us-east-1-497443303796&prefix=. The main area displays a table of objects in the 'elasticbeanstalk-us-east-1-497443303796' bucket. The columns are Name, Storage Class, Size, and Last Modified. The table lists numerous files and folders, mostly named after Java WAR files from various dates. A search bar at the top right allows filtering by prefix. Below the table, there are links for Feedback, English, Privacy Policy, and Terms of Use, along with system status icons like battery level and network connection.

Name	Storage Class	Size	Last Modified
.elasticbeanstalk	Standard	0 bytes	Wed Nov 16 21:50:24 GMT-500 2016
2016322mDC-FinalProject.war	Standard	15.3 MB	Wed Nov 16 21:48:30 GMT-500 2016
2016325IJx-FinalProject.war	Standard	15.3 MB	Sun Nov 20 12:45:10 GMT-500 2016
2016328FcN-FinalProject.war	Standard	15.3 MB	Wed Nov 23 02:41:43 GMT-500 2016
2016328Pm-FinalProject.war	Standard	15.3 MB	Wed Nov 23 14:58:10 GMT-500 2016
2016340z08-CloudProject.war	Standard	15.3 MB	Mon Dec 05 13:54:57 GMT-500 2016
2016343Kl6-CloudProject.war	Standard	62.5 MB	Thu Dec 08 07:28:34 GMT-500 2016
2016343L4T-CloudProject.war	Standard	62.5 MB	Thu Dec 08 10:53:11 GMT-500 2016
2016343Y7Q-FinalProject2.war	Standard	15.3 MB	Thu Dec 08 14:50:15 GMT-500 2016
2016343xtY-CloudProject.war	Standard	62.5 MB	Thu Dec 08 07:46:42 GMT-500 2016
2016349CKH-s3.war	Standard	62.5 MB	Tue Dec 13 20:31:45 GMT-500 2016
2016349Fvt-s3.war	Standard	62.5 MB	Wed Dec 14 16:05:41 GMT-500 2016
2016349qBs-s3.war	Standard	62.5 MB	Wed Dec 14 02:26:48 GMT-500 2016
2016350e5P-s3.war	Standard	62.5 MB	Wed Dec 14 19:54:25 GMT-500 2016
2016350nBD-s3.war	Standard	62.5 MB	Wed Dec 14 20:05:19 GMT-500 2016
resources	-	--	--

# Elastic IP attached to the EC2 instance

The screenshot shows the AWS EC2 Management console interface. The left sidebar navigation bar is visible, with the 'Elastic IPs' option highlighted. The main content area displays a table of allocated Elastic IPs. One row is selected, showing details for an Elastic IP with the address 34.194.157.81. The selected row includes columns for Allocation ID, Instance, Private IP Address, Scope, and Public DNS. Below this table, a detailed view for the selected Elastic IP is shown, listing its association with an EC2 instance (i-090043271530ac947) and its network interface information (eni-eb08fc1a).

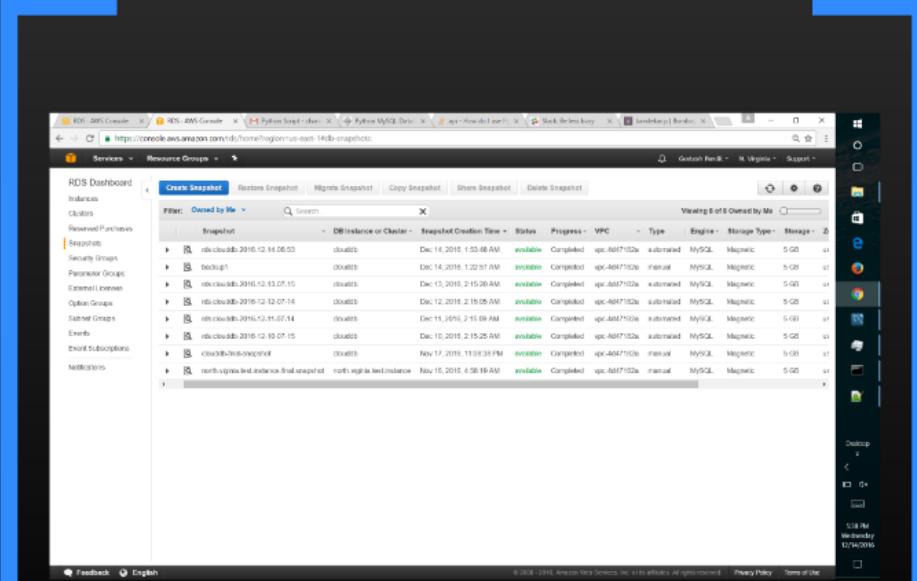
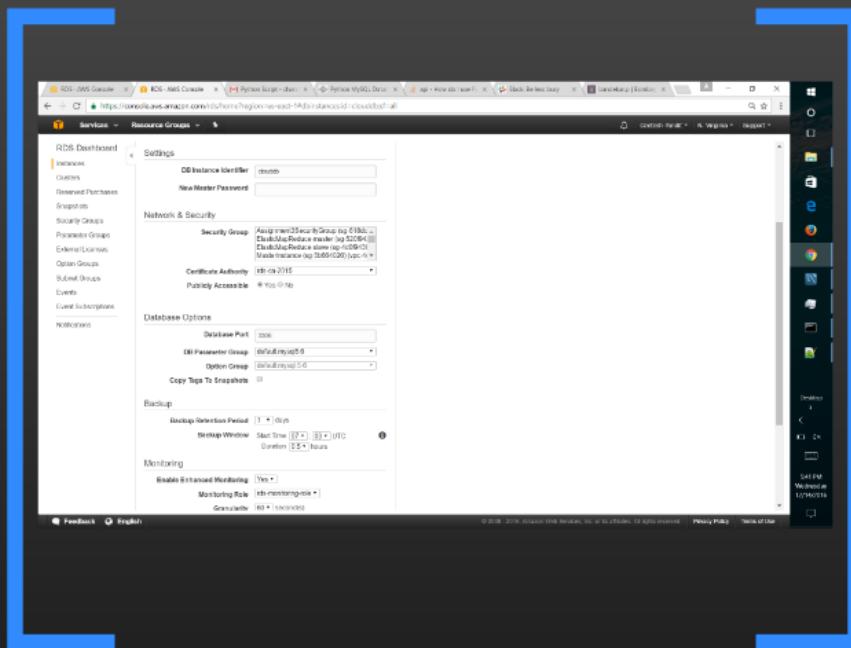
Elastic IP	Allocation ID	Instance	Private IP Address	Scope	Public DNS
34.194.157.81	eipalloc-43ea167d	i-090043271530ac947 (LowCost-env)	172.31.51.165	vpc	ec2-34-194-157-81.compute-1.amazonaws.com

**Address: 34.194.157.81**

Elastic IP	34.194.157.81	Network interface ID	eni-eb08fc1a
Instance	i-090043271530ac947 (LowCost-env)	Private IP address	172.31.51.165
Scope	vpc	Network interface owner	497443303796
Public DNS	ec2-34-194-157-81.compute-1.amazonaws.com	Allocation ID	eipalloc-43ea167d

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use 11:11 PM 12/14/2016

# Snapshots of the RDS Db is taken every day.



RDS · AWS Console x RDS · AWS Console x Python Script - chan... x Python MySQL Data... x api - How do I use Py... x Slack: Be less busy x # bandekar.p | Bombay x

https://console.aws.amazon.com/rds/home?region=us-east-1#dbinstances:id=clouddb;sf=all

Services Resource Groups

RDS Dashboard

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

Settings

DB Instance Identifier: clouddb

New Master Password: [redacted]

Network & Security

Security Group: Assignment3SecurityGroup (sg-618dc...) ▾  
ElasticMapReduce-master (sg-520f94:...)  
ElasticMapReduce-slave (sg-4d0f943(...)  
MasterInstance (sg-5b664026) (vpc-4c...)

Certificate Authority: rds-ca-2015

Publicly Accessible: Yes

Database Options

Database Port: 3306

DB Parameter Group: default.mysql5.6

Option Group: default:mysql-5.6

Copy Tags To Snapshots:

Backup

Backup Retention Period: 1 days

Backup Window: Start Time: 07 : 03 UTC Duration: 0.5 hours

Monitoring

Enable Enhanced Monitoring: Yes

Monitoring Role: rds-monitoring-role

Granularity: 60 second(s)

Feedback English

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

Privacy Policy Terms of Use

5:41 PM Wednesday 12/14/2016

Desktop

File Explorer Task View Taskbar

RDS - AWS Console x RDS - AWS Console x Python Script - chan... x Python MySQL Data... x api - How do I use Py... x Slack: Be less busy x # bandekar.p | Bombay x

https://console.aws.amazon.com/rds/home?region=us-east-1#db-snapshots:

RDS Dashboard

Instances Clusters Reserved Purchases Snapshots Security Groups Parameter Groups External Licenses Option Groups Subnet Groups Events Event Subscriptions Notifications

Create Snapshot Restore Snapshot Migrate Snapshot Copy Snapshot Share Snapshot Delete Snapshot

Filter: Owned by Me Search... Viewing 8 of 8 Owned by Me

Snapshot	DB Instance or Cluster	Snapshot Creation Time	Status	Progress	VPC	Type	Engine	Storage Type	Storage	Zone
rds.clouddb-2016-12-14-06-53	clouddb	Dec 14, 2016, 1:53:48 AM	available	Completed	vpc-4d47182a	automated	MySQL	Magnetic	5 GB	us
backup1	clouddb	Dec 14, 2016, 1:22:57 AM	available	Completed	vpc-4d47182a	manual	MySQL	Magnetic	5 GB	us
rds.clouddb-2016-12-13-07-15	clouddb	Dec 13, 2016, 2:15:20 AM	available	Completed	vpc-4d47182a	automated	MySQL	Magnetic	5 GB	us
rds.clouddb-2016-12-12-07-14	clouddb	Dec 12, 2016, 2:15:05 AM	available	Completed	vpc-4d47182a	automated	MySQL	Magnetic	5 GB	us
rds.clouddb-2016-12-11-07-14	clouddb	Dec 11, 2016, 2:15:09 AM	available	Completed	vpc-4d47182a	automated	MySQL	Magnetic	5 GB	us
rds.clouddb-2016-12-10-07-15	clouddb	Dec 10, 2016, 2:15:25 AM	available	Completed	vpc-4d47182a	automated	MySQL	Magnetic	5 GB	us
clouddb-final-snapshot	clouddb	Nov 17, 2016, 11:08:38 PM	available	Completed	vpc-4d47182a	manual	MySQL	Magnetic	5 GB	us
north-virginia-test-instance-final-snapshot	north-virginia-test-instance	Nov 16, 2016, 4:58:19 AM	available	Completed	vpc-4d47182a	manual	MySQL	Magnetic	5 GB	us

Feedback English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Windows O e e Desktop < > 5:38 PM Wednesday 12/14/2016

# Multi-Availability zone

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar navigation bar includes links for Services (selected), Resource Groups, IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE (Volumes, Snapshots), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), LOAD BALANCING (selected, showing Load Balancers and Target Groups), and AUTO SCALING (Launch Configurations). The main content area displays a table of existing load balancers. One row is selected, showing details for a load balancer named "awseb-e-n-AWSEBLoa-1NG..." with a DNS name of "awseb-e-n-AWSEBLoa-1NG...". It is associated with VPC ID "vpc-4d47182a" and is located in availability zones "us-east-1a, us-east-1b, us-east-1d". The status is "InService". Below this, an "Edit Availability Zones" section lists three subnets: subnet-4599410c (us-east-1a), subnet-701cf32b (us-east-1b), and subnet-a009e08d (us-east-1d). Each subnet has an instance count of 3 and is marked as healthy. Actions for each subnet include "Remove from Load Balancer". The bottom of the page includes standard AWS footer links for Feedback, English, Privacy Policy, Terms of Use, and a "Ask me anything" button.

Name	DNS name	State	VPC ID	Availability Zones
awseb-e-n-AWSEBLoa-1NG...	awseb-e-n-AWSEBLoa-1NG...	InService	vpc-4d47182a	us-east-1a, us-east-1b, us-east-1d

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-east-1a	subnet-4599410c	172.31.0.0/20	4	Yes	Remove from Load Balancer
us-east-1b	subnet-701cf32b	172.31.16.0/20	3	Yes	Remove from Load Balancer
us-east-1d	subnet-a009e08d	172.31.48.0/20	3	Yes	Remove from Load Balancer



A large, stylized, blue cutout sign with a white outline and a drop shadow, reading "thank you". The sign is suspended by a thin brown string from a small metal hook at the top center. The background is plain white.

thank  
you

# Cloud Computing Project

Priyanka Bandekar  
Maansi Chandira  
Geetesh Pandit  
Siddharth Wale

## Requirements

1. Develop a login portal using Java Spring which will consist of user-case buttons to simulate an increasing load on the application and database.
2. The database can either be a relational database or a NoSQL database.
3. The application needs to be deployed on AWS.
4. Proper re-hausurcement and design to allow for auto-scaling of resources to accommodate the additional load in application, network, data storage and compute.
5. Minimum of one load balancer is required.
6. There are 5 use-case buttons that need to be implemented which are:
  - a. Simulation of 3 user login to application and run 2 report going back 10 days
  - b. Simulation of 10 user login to application and run 6 report going back 30 days
  - c. Simulation of 17 user login to application and run 10 report going back 60 days
  - d. Simulation of 24 user login to application and run 14 report going back 90 days
  - e. User input fields to allow me the ability to enter parameters for how many users are simulated and how many reports those users are querying site running.

## Auto Scaling

In our project we have used AWS CloudWatch which handles the auto scaling feature. If any of the EC2 instances fail to respond, the CloudWatch detect it and launch a new instance. Other auto scale and cloud metrics can be terminated automatically.



## CloudWatch

In our project we have used Amazon CloudWatch to monitor AWS resources such as Amazon EC2 instances and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate.



Route to Load Balancer  
  
In the load balancer we  
are going to redirect all the HTTP



## Business Justification

We are in the business of renting cars to help people go wherever they want.  
Demand for our cars varies largely according to different factors such as Day of week, Long Weekends and sometimes even on Happy Hours of day.

To satisfy our varying demands it is crucial for us to have a scalable architecture which could scale up and scale down as per business needs.

AWS provides a simple and cost efficient way to host our infrastructure on cloud.



## 6225-CSYE Network Structures and Cloud Computing

## Web Application

We have created a Car Rental Web Application where Customers can register and create an account. After creating an account, Customer can login to the portal and reserve data for him.

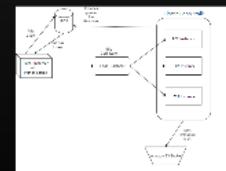
A report page is fetching data about the Cars from the database. Besides the main report, we have a mechanism to simulate users.

## User Simulation

We are using a python script to create a web client, which will open a socket connection to the server and request a web page from the application. When we use any of the buttons on the report page to simulate user requests to the application, the web application creates a process to execute the python script. Depending on the value of user count being simulated, the script opens that many connections to the server and tries to fetch data.



## Project Architecture



- Our infrastructure has been built allowing for having a master and redundancy. It is on standby.  
- We typically have some pre-configured Amazon Machine Images (AMIs)  
- Use Elastic IP addresses, which are identified in the preparation phase with them with your instances.  
- Use Elastic Load Balancing (ELB) multiple instances.  
- An Amazon EBS snapshot to ensure installation packages and configurations in AWS.