

CS55A course project presentation



Wild Rydes platform on AWS

By: Yilin Shen

Components of the platform and how it designed

Systems Architect use AWS S3, Amazon Cognito, AWS Lambda, Amazon DynamoDB, Amazon API Gateway to implement our website.

Static Web Hosting

The AWS S3 holds all the files and packages such as JavaScript, HTML, CSS, image of the static web site. We can access WildRydes web site by using the URL it provide.

User Management

Amazon Cognito let us manage which user can access our website at the backend to secure API.

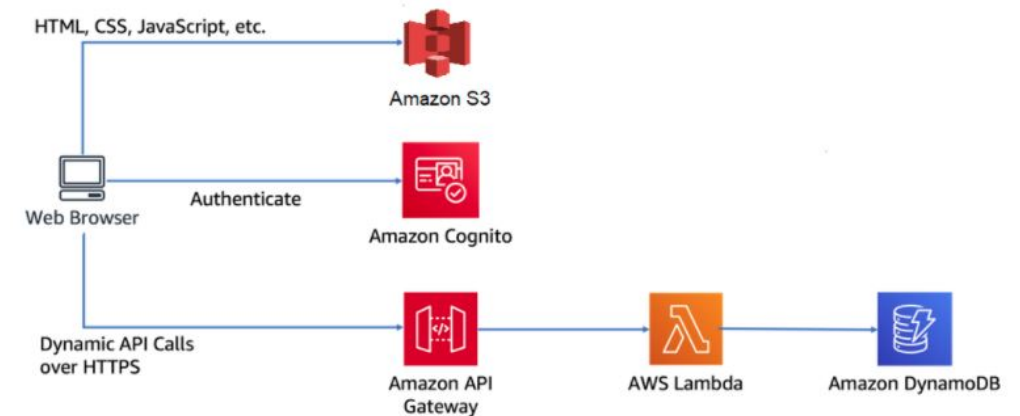
Serverless Backend

Amazon Lambda plays the role to run the code to conduct the API request from the platform application. The requestUnicorn.js code can implement dispatch a unicorn on the web site as the new Lambda function.

After the fuction selected a unicorn from the fleet, DynamoDB stored the request record. IAM role grants the Lambda function permission to write logs to Amazon CloudWatch Logs and access to write items to your DynamoDB table.

RESTful API

Amazon API Gateway can make the Lambda function as a RESTful API. It will be secured using the Amazon Cognito. This will also turn the statically hosted website into a dynamic web application.



Systems Architect structure chart

Amazon Cognito role in platform implementation?

The screenshot shows the Amazon Cognito console interface. On the left, there's a sidebar with navigation options: General settings, Users and groups (highlighted), App integration, and Federation. The main area displays the 'Users' tab for a user pool named 'WildRydes'. At the top of the main area, there are buttons for 'Import users' and 'Create user', and a search bar labeled 'User name' with a dropdown arrow and a placeholder 'Search for value.'. Below this is a table with the following columns: Username, Enabled, Account status, Email, and Email verified. The table contains six rows of user data.

Username	Enabled	Account status	Email	Email verified
1441903534@pp.com	Enabled	CONFIRMED	1441903534@pp.com	false
yilin.shen92@gmail.com	Enabled	UNCONFIRMED	yilin.shen92@gmail.com	false
yilinshe@usc.edu	Enabled	CONFIRMED	yilinshe@usc.edu	true
yilinshen.92@gmail.com	Enabled	UNCONFIRMED	yilinshen.92@gmail.com	false
yilinshen1992@gmail.com	Enabled	CONFIRMED	yilinshen1992@gmail.com	false

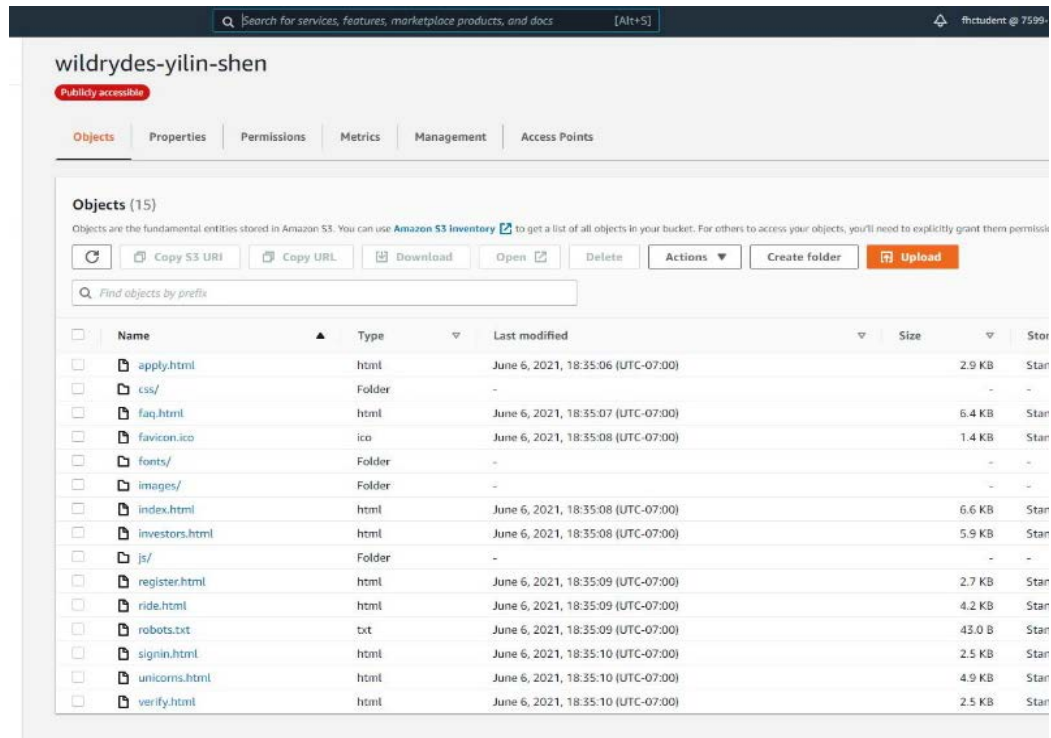
Users requested to register

- Amazon Cognito user pool can confirm the user account status. If the user use a real email address and verify by the verification code he/she can sign in.
- If the user use a dummy email address, we have to confirm the user manually through the Cognito console.
- If the user didn't successfully complete registration, he/she can't login and go to the next map section.

Amazon S3 roles for the platform

Amazon S3 has the bucket holds all the static web content, such as html files, codes files, css files, txt files, images, packages and all other find of information.

The URL under S3 properties static website hosting allow access to the public website.



Amazon S3 bucket holds objects

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://wildrydes-yilin-shen.s3-website-us-west-1.amazonaws.com>

URL to the public website

Amazon S3 roles for the platform – bucket policy

We using buckets policy to define who can access the bucket. The code in screenshots provide the bucket name that will be allow to access. JSON documents specify what principals are allowed to execute various actions against the objects in bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::[YOUR_BUCKET_NAME]/*"
    }
  ]
}
```

screenshot JSON documents

What role does Amazon Lambda play in the platform?
What does the code do?

Amazon Lambda plays the role to run the code to conduct the API request from the platform application.

The requestUnicorn.js code can implement dispatch a unicorn on the web site as the new Lambda function.

Lambda function for requesting a Unicorn to come pick us up. Part of the WildRydes AWS application

```
requestUnicorn.js

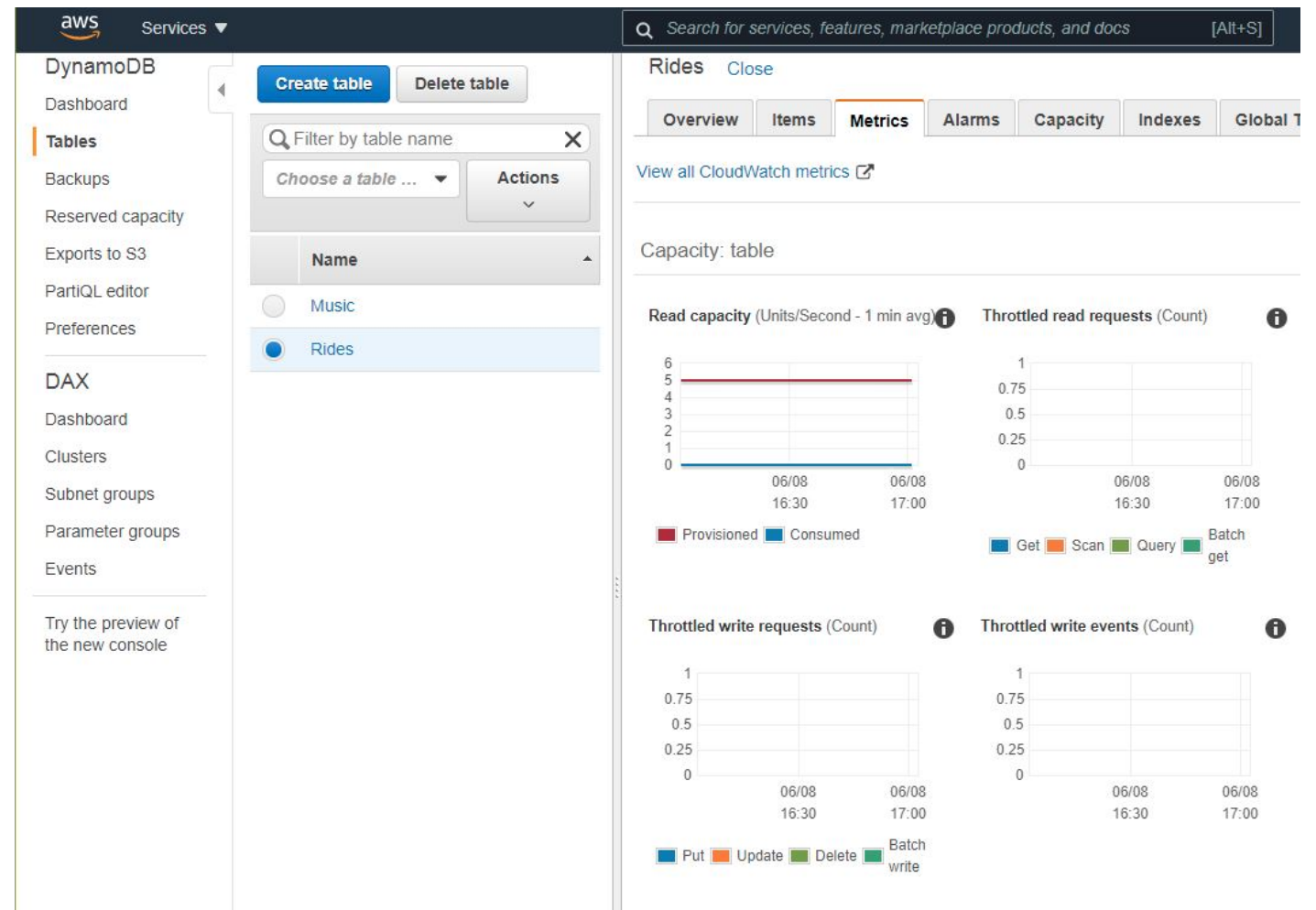
1  const randomBytes = require('crypto').randomBytes;
2
3  const AWS = require('aws-sdk');
4
5  const ddb = new AWS.DynamoDB.DocumentClient();
6
7  const fleet = [
8    {
9      Name: 'Bucephalus',
10     Color: 'Golden',
11     Gender: 'Male',
12   },
13   {
14     Name: 'Shadowfax',
15     Color: 'White',
16     Gender: 'Male',
17   },
18   {
19     Name: 'Rocinante',
20     Color: 'Yellow',
21     Gender: 'Female',
22   },
23 ];
24
25 exports.handler = (event, context, callback) => {
26   if (!event.requestContext.authorizer) {
27     errorResponse('Authorization not configured', context.awsRequestId, callback);
28     return;
29   }
```

part of requestUnicorn.js

What Amazon DynamoDB store & IAM role grant

After the fuction selected a unicorn from the fleet, DynamoDB stored the request record.

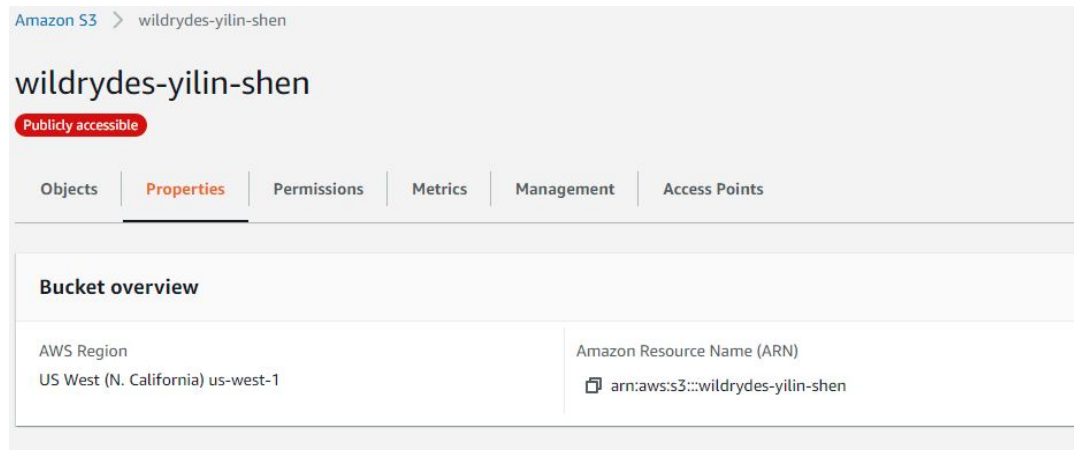
IAM role grants the Lambda function permission to write logs to Amazon CloudWatch Logs and access to write items to your DynamoDB table.



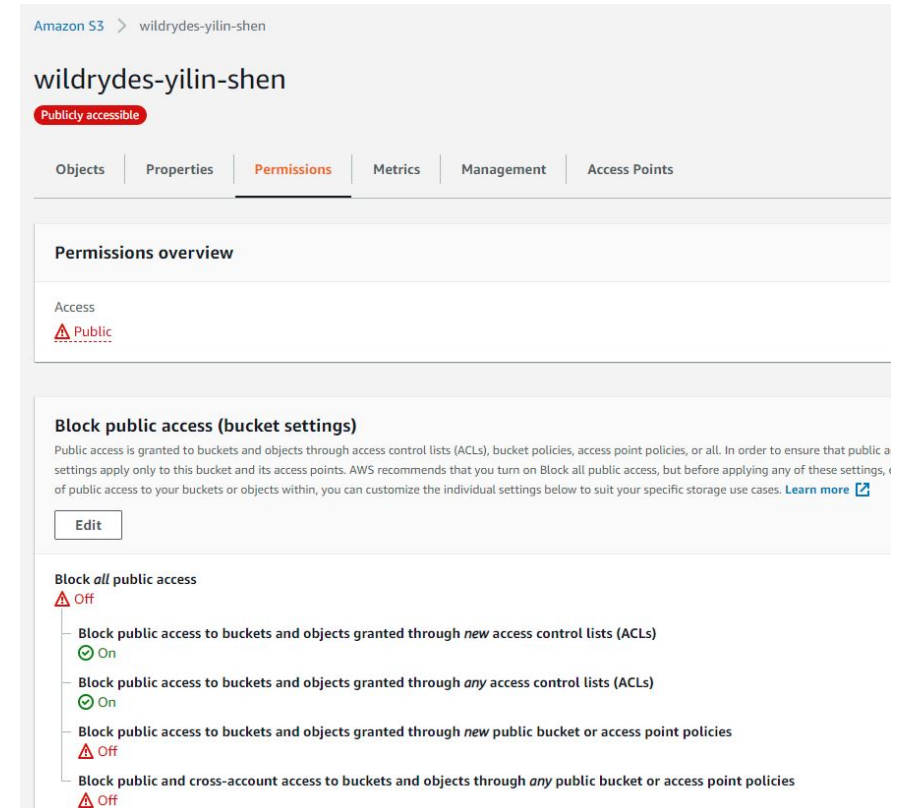
Amazon CloudWatch metrics, alarms under the DynamoDB

Design Choices made along the way

- Selected US West (N.California) us-west-1 region. Because most of the design base and unicorn call service will be in Bay area.
- In Public access setting, only blocked "Block new public bucket policies" and "Block public and cross-account access if bucket has public policies", since we need to let anonymous users view our site. If block all of them, only authenticated users with access to AWS account



US West (N.California) us-west-1 region



Public access setting

Components including & validations

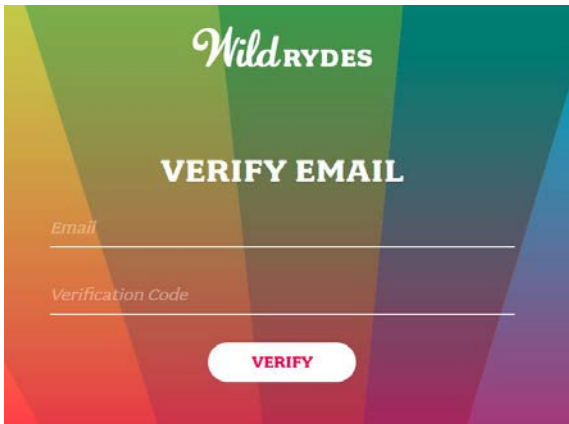
In the home page under words Wild RYDES user can select GIDDY UP! to register the email address and confirm password. Then goes to verify email inter verification code. After verify user can sign in to use the call unicorn section



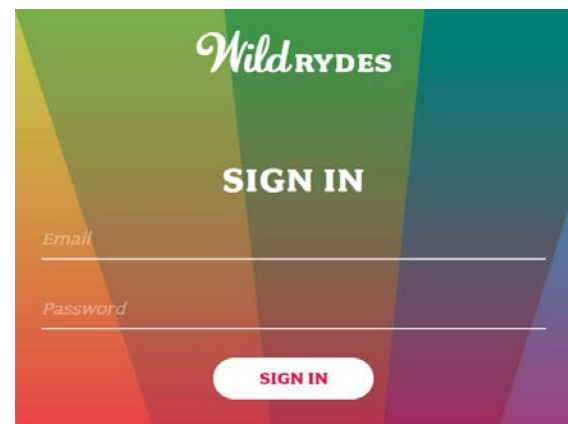
home page



Register page

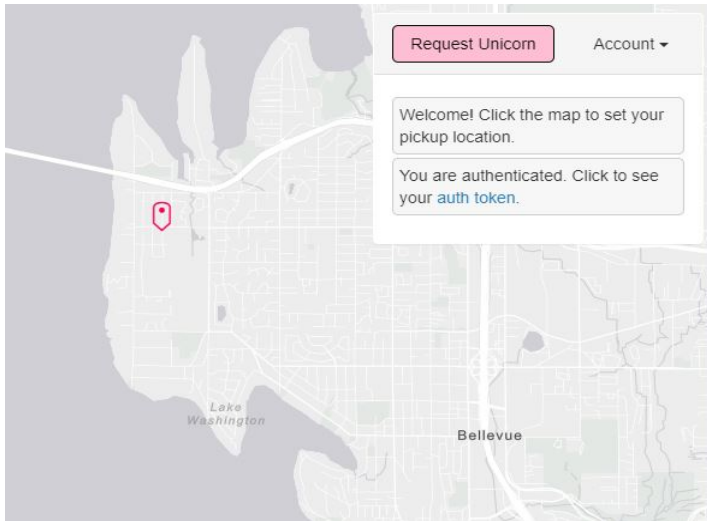


Verify page

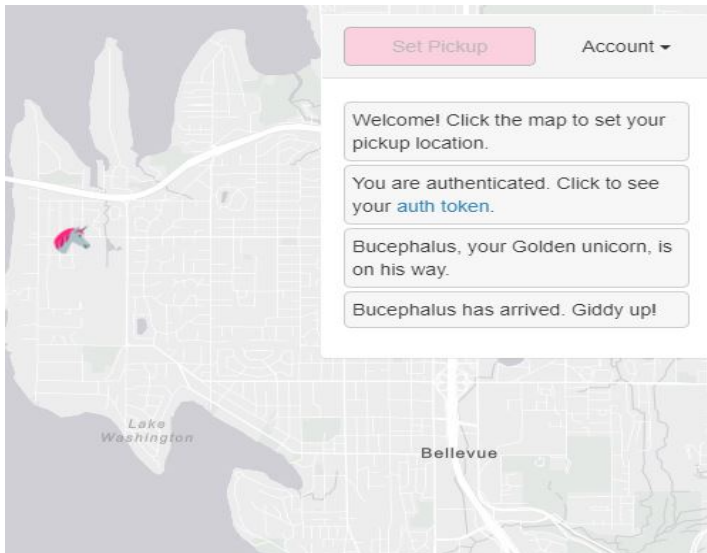


sign in page

Components including & validations



request unicorn map section



Unicorn arrived map section

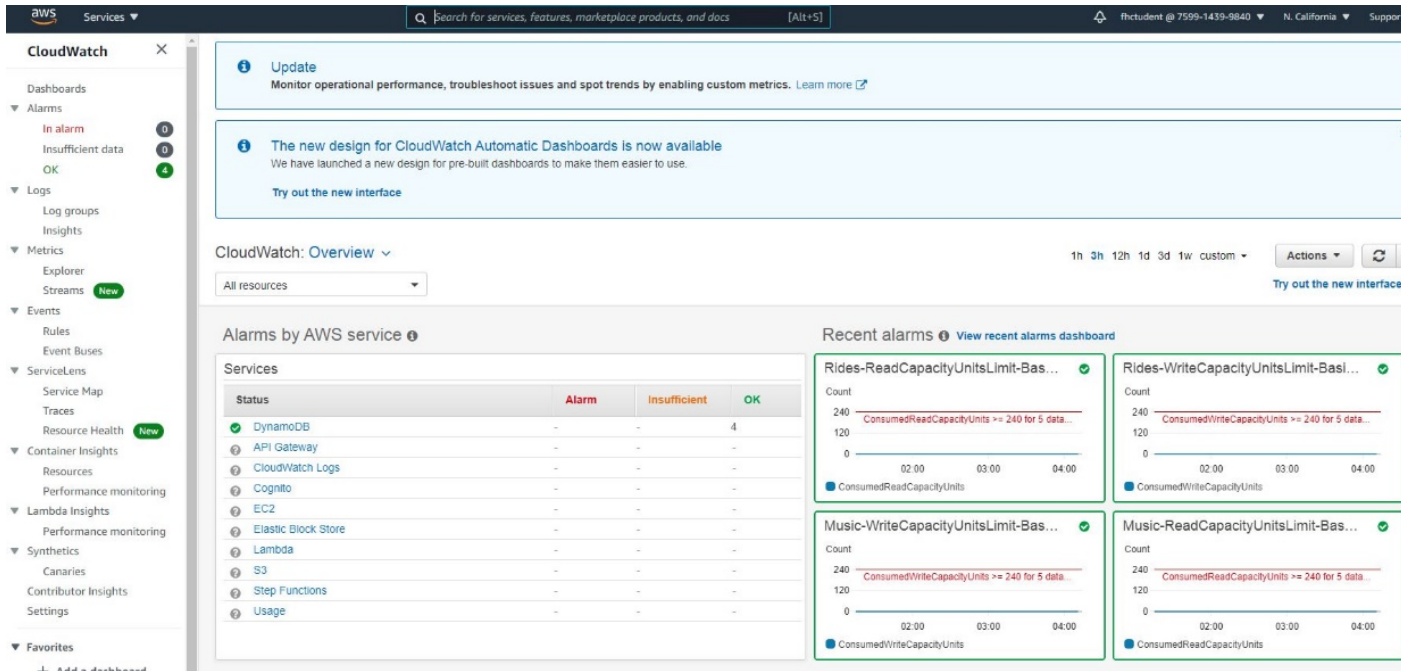
- customer first set the Pick up location then request unicorn to come.



section directions

- after the request, the notification shows that a unicorn is on the way with its name and color. When the unicorn arrives the notification pump up.
- click the right up corner of the home page. Users can decide which sections to goes to.

Costs of running the WildRydes platform



CloudWatch page

We can use Cloud Watch to track the usage. But the cost analysis will be based on assumption. Customers will be in N. California area and grow from 1-100 to 1000 to 10000 per second. We don't offer unicorn pool, so one unicorn ride will be requested by one registered user.

Using Simple Monthly Calculator

Services			Estimate of your Monthly Bill (\$ 41.84)
Estimate of Your Monthly Bill <input type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)			
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.			
Export to CSV			Save and Share
<input type="checkbox"/>	Amazon EC2 Service (US West (Northern California))		\$ 19.71
<input type="checkbox"/>	Amazon S3 Service (US West (Northern California))		\$ 0.16
<input type="checkbox"/>	Amazon Elastic Load Balancing (US West (Northern California))		\$ 20.62
<input type="checkbox"/>	Amazon DynamoDB Service (US West (Northern California))		\$ 0.00
<input type="checkbox"/>	Amazon SNS Service (US West (Northern California))		\$ 0.00
<input type="checkbox"/>	Amazon SQS Service (US West (Northern California))		\$ 0.00
<input type="checkbox"/>	AWS Data Transfer In		\$ 0.00
<input type="checkbox"/>	AWS Data Transfer Out		\$ 1.35
<input type="checkbox"/>	AWS Support (Basic)		\$ 0.00
Total Monthly Payment:			\$ 41.84

Elastic Bean Stalk usage monthly cost

<input type="checkbox"/> FREE TIER: New Customers get free usage tier for first 12 months			
Services			Estimate of your Monthly Bill (\$ 213.44)
Estimate of Your Monthly Bill <input checked="" type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)			
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.			
Export to CSV			Save and Share
<input type="checkbox"/>	Amazon EC2 Service (US West (Northern California))		\$ 0.00
<input type="checkbox"/>	Amazon S3 Service (US West (Northern California))		\$ 59.83
<input type="checkbox"/>	Amazon CloudFront Service		\$ 133.00
<input type="checkbox"/>	AWS Data Transfer In		\$ 0.00
<input type="checkbox"/>	AWS Data Transfer Out		\$ 20.61
<input type="checkbox"/>	AWS Support (Basic)		\$ 0.00
Total Monthly Payment:			\$ 213.44

Marketing Web Site monthly cost

- Later on the company will move to Marketing Web Site with larger data transfer to 30GB. The monthly cost will be \$213.44

- WildRydes won't get free tiers so at the very beginning, We will use Elastic Bean Stalk which allows 1 Micro Instance, Running 24 Hours/Day, 15 GB of Data Transfer. Handling "300" requests/moth concurrently. The cost per month is \$41.84

Services			Estimate of your Monthly Bill (\$ 213.44)								
Choose region: US West (Northern California)			Inbound Data Transfer is Free and Outbound Data Transfer is \$0.12/GB								
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.											
Newer versions of the EC2 calculators are available: Amazon EC2 , EC2 Dedicated Host , Elastic Graphics , Elastic IP											
Compute: Amazon EC2 Instances:											
	Description	Instances	Usage	Type	Billing Option	Monthly Cost					
<input type="button" value="Add New Row"/>											
Compute: Amazon EC2 Dedicated Hosts:											
	Description	Number of Hosts	Usage	Type	Billing Option						
<input type="button" value="Add New Row"/>											
Storage: Amazon EBS Volumes:											
	Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput	Snapshot Storage				
<input type="button" value="Add New Row"/>											
Additional T2/T3/T4g Unlimited vCPU Hours per month:											
T4g For Linux, RHEL and SLES:			<input type="text" value="0"/>								
For Linux, RHEL and SLES:			<input type="text" value="0"/>								
For Windows and Windows with SQL Web:			<input type="text" value="0"/>								
Elastic IP:											
<input checked="" type="radio"/> Enter values below <input type="radio"/> Calculate											
Total time the additional Elastic IPs are attached to running EC2 instances**:			<input type="text" value="0"/>	Hours/Month							
Total Non-attached time for all the Elastic IPs:			<input type="text" value="0"/>	Hours/Month							
Number of Elastic IP Remaps:			<input type="text" value="0"/>	Per Month							
Data Transfer:											
Inter-Region Data Transfer Out:			<input type="text" value="0"/>	GB/Month							
Data Transfer Out:			<input type="text" value="30"/>	GB/Month							
Data Transfer In:			<input type="text" value="20"/>	GB/Week							
VPC Peering Data Transfer:			<input type="text" value="0"/>	GB/Month							
Intra-Region Data Transfer:			<input type="text" value="0"/>	GB/Month							
Public IP/Elastic IP Data			<input type="text" value="0"/>	GB/Month							

Marketing Web site capacity

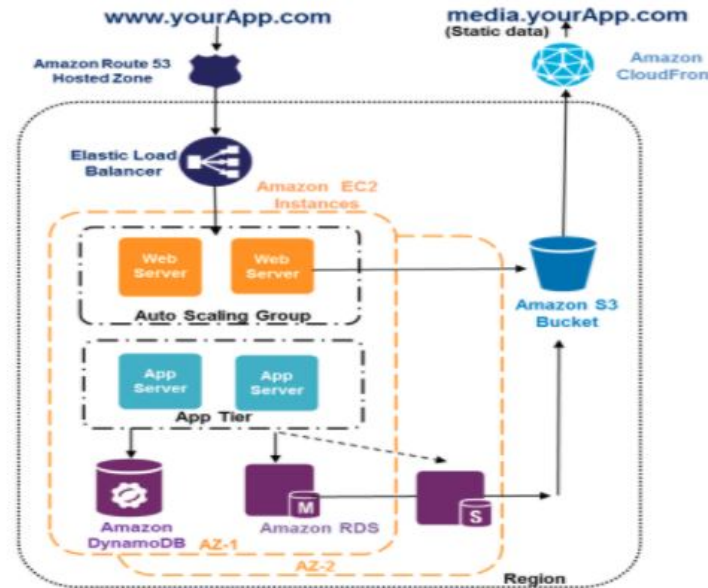
Using Simple Monthly Calculator

With the growth of the customers, platform will move to large web application. This will provide the company 1 Load Balancer, 2 Web serves, 2 App Serves, 1 High Availability database server, 30 GB of storage, and 120 BG of data transfer.

The cost per on the will be \$1127.18

AWS Solution Calculator

Monthly Costs of Deploying this Solution on AWS: \$ 1095.73



Name: 3-Tier Auto-scalable Web Application Solution

Includes:

1 Load Balancer, 2 Web Servers, 2 App Servers, 1 High Availability Database Server, 30 GB of Storage; 120 GB of Data Transfer

Description:

Ruby on Rails application can serve approximately 100,000 pageviews per month

Services

Estimate of your Monthly Bill (\$ 1127.18)

Estimate of Your Monthly Bill

☐ Show First Month's Bill (include all one-time fees, if any)

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.

Export to CSV

Save and Share

1	Amazon EC2 Service (US West (Northern California))	\$	645.58
2	Amazon S3 Service (US West (Northern California))	\$	0.89
3	Amazon Route 53 Service	\$	0.90
4	Amazon CloudFront Service	\$	34.16
5	Amazon RDS Service (US West (Northern California))	\$	291.00
6	Amazon Elastic Load Balancing (US West (Northern California))	\$	22.22
7	Amazon DynamoDB Service (US West (Northern California))	\$	90.22
8	AWS Data Transfer In	\$	0.00
9	AWS Data Transfer Out	\$	42.21
10	AWS Support (Basic)	\$	0.00
Total Monthly Payment:		\$	1127.18

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

large Web Application monthly cost

AWS pricing calculator- REST APIs

Using AWS pricing calculator, we can estimate the cost of REST APIs per month. When the data is under 100 request/s the monthly cost will be around 400 USD. As the request amount goes up to 1000/s the cost goes up to 745 USD. For a 10000 request/s usage the monthly cost is 4080.99 USD.

The screenshot shows the AWS Pricing Calculator interface for REST APIs. The 'Requests' field is set to 108, and the 'Cache memory size (GB)' is set to 28.4. The 'Show calculations' section displays the following breakdown:

- 108 requests x 1,000,000 unit multiplier = 108,000,000 total REST API requests
- Tiered price for: 108000000 requests
- 108000000 requests x 0.0000035000 USD = 378.00 USD
- Total tier cost = 378.0000 USD (REST API requests)
- Tiered price total for REST API requests: 378 USD**
- 0.50 USD per hour x 730 hours in a month = 365.00 USD for cache memory
- Dedicated cache memory total price: 365.00 USD**
- 378 USD + 365.00 USD = 743.00 USD
- REST API cost (monthly): 743.00 USD**

Up to 1000 request/s monthly cost

The screenshot shows the AWS Pricing Calculator interface for REST APIs. The 'Requests' field is set to 1080, and the 'Cache memory size (GB)' is set to 28.4. The 'Show calculations' section displays the following breakdown:

- 1,080 requests x 1,000,000 unit multiplier = 1,080,000,000 total REST API requests
- Tiered price for: 1080000000 requests
- 333000000 requests x 0.0000035000 USD = 1165.50 USD
- 667000000 requests x 0.0000034700 USD = 2314.49 USD
- 80000000 requests x 0.0000029500 USD = 236.00 USD
- Total tier cost: 1165.50 USD + 2314.49 USD + 236.00 USD = 3715.99 USD (REST API requests)
- Tiered price total for REST API requests: 3,715.99 USD**
- 0.50 USD per hour x 730 hours in a month = 365.00 USD for cache memory
- Dedicated cache memory total price: 365.00 USD**
- 3,715.99 USD + 365.00 USD = 4,080.99 USD
- REST API cost (monthly): 4,080.99 USD**

Up to 10000 request/s monthly cost

The screenshot shows the AWS Pricing Calculator interface for REST APIs. The 'Requests' field is set to 10, and the 'Cache memory size (GB)' is set to 28.4. The 'Show calculations' section displays the following breakdown:

- 10 requests x 1,000,000 unit multiplier = 10,000,000 total REST API requests
- Tiered price for: 10000000 requests
- 10000000 requests x 0.0000035000 USD = 35.00 USD
- Total tier cost = 35.0000 USD (REST API requests)
- Tiered price total for REST API requests: 35 USD**
- 0.50 USD per hour x 730 hours in a month = 365.00 USD for cache memory
- Dedicated cache memory total price: 365.00 USD**
- 35 USD + 365.00 USD = 400.00 USD
- REST API cost (monthly): 400.00 USD**

1-100 request/s monthly cost