

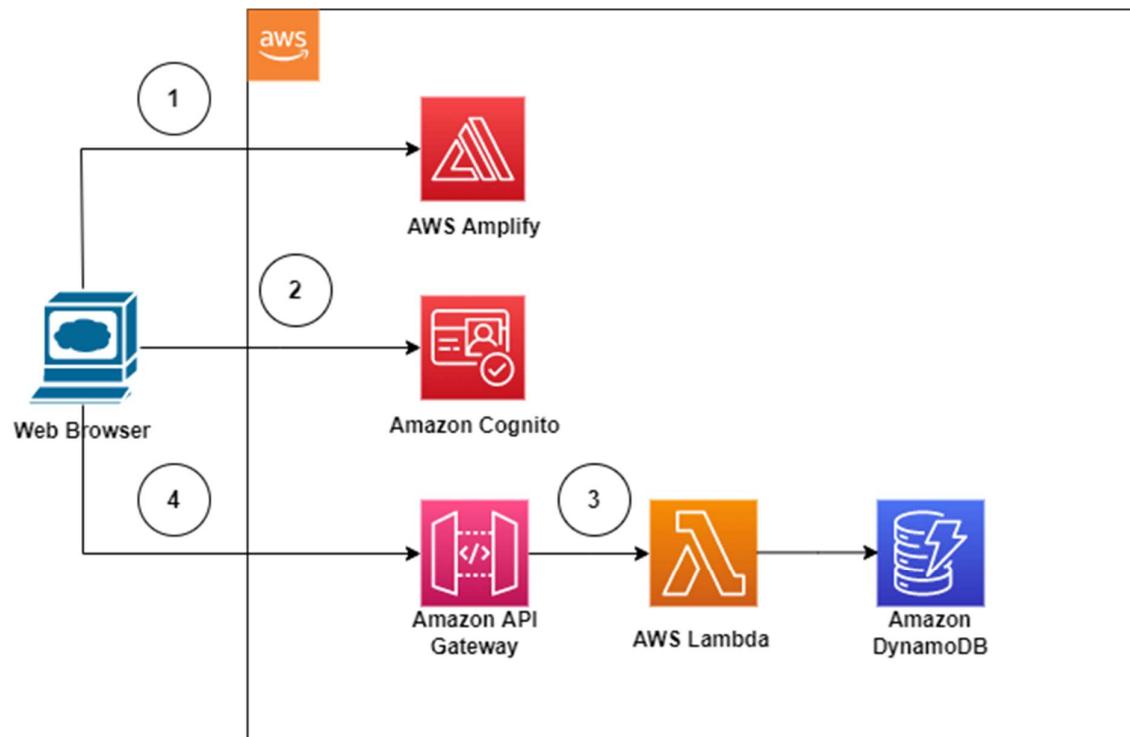
Build a serverless web application using Amplify, Cognito, API Gateway, Lambda & DynamoDB

Project Overview: -

Create a simple serverless web application that enables users to request unicorn rides from Wild Rydes fleet. Application will present a HTML based user interface for users to register with the service and login. Users would then be able to navigate to unicorn ride page where they can select the location they would like to be picked up and will interface on the backend with RESTful API to submit the request and dispatch a nearby unicorn.

Architecture: -

Application architecture uses AWS Lambda, Amazon API Gateway, Amazon DynamoDB, Amazon Cognito, and AWS Amplify Console. Amplify Console provides continuous deployment and hosting of the static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser. JavaScript executed in the browser sends and receives data from a RESTful API built using Lambda and API Gateway. Amazon Cognito provides user management and authentication functions to secure the backend API. Finally, DynamoDB provides a persistence layer where data can be stored by the API's Lambda function.



Serverless Web Application Architecture Diagram using AWS Managed resources

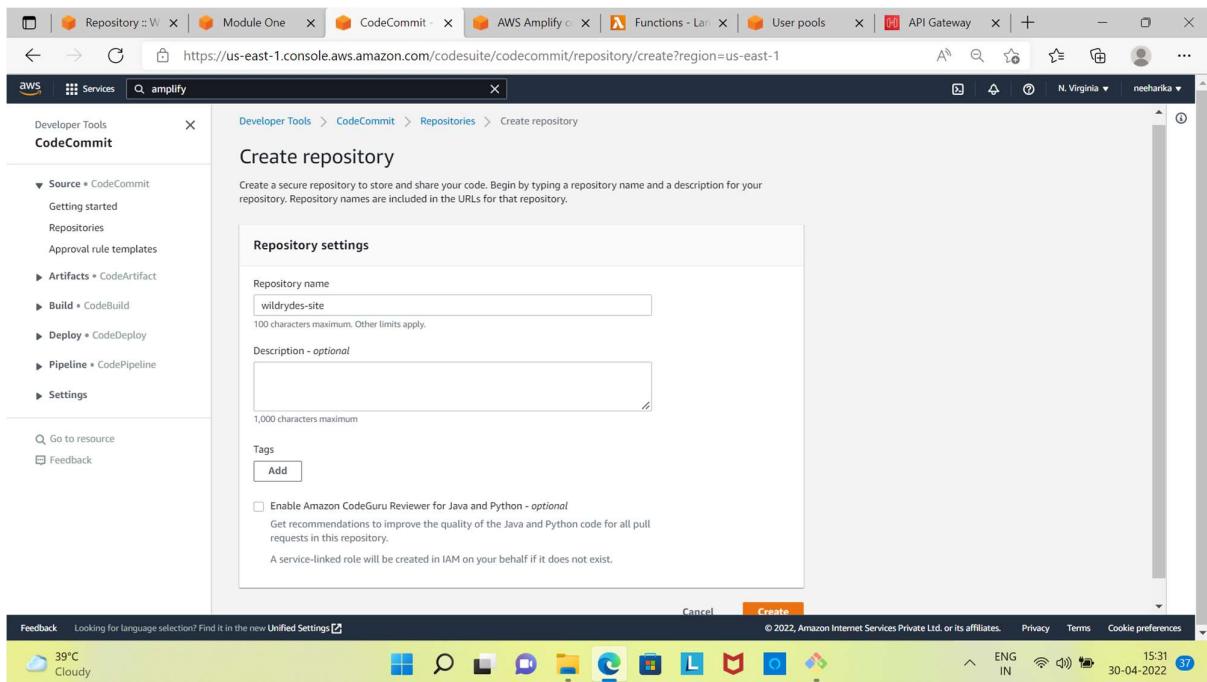
Architecture Implementation	
1	Static Web Hosting - AWS Amplify hosts static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser.
2	User Management - Amazon Cognito provides user management and authentication functions to secure backend API i.e, Sends a set of JSON web tokens (JWT).
3	Serverless Backend – Lambda Function invoked from browser using Amazon API gateway. Function responds to front end application and records the request in DynamoDB Table
4	REST API – Dynamic API calls over HTTPS. JWT sent by Cognito contains identity of the user and used to authenticate REST API build with Amazon API Gateway.

Note: Entire project has been developed using AWS account in us-east-1 region.

Project Implementation – Instructions with Snapshots: -

Step1: Static Web Hosting

1. Create Git repository using AWS Code commit and clone it locally.



2. Populate Git repository - Copy the web site content from an existing publicly accessible S3 bucket associated with wild rydes and add the content to your repository.

```
MINGW64:c/Users/heeha/Desktop/serverless/wildrydes-site
heeha@DESKTOP-9EGL690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ aws s3 cp s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website ./ --recursive
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/apply.html to \.apply.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/index.css to css\index.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/font.css to css\font.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/message.css to css\message.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/ride.css to css\ride.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/normalize.css to css\normalize.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/main.css to css\main.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/faq.html to \faq.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/favicon.ico to \favicon.ico
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/fairplex-wide-n7.woff to fonts\fairplex-wide-n7.woff
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/glyphicons-halflings-regular.woff to fonts\glyphicons-halflings-regular.woff
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/glyphicons-halflings-regular.ttf to fonts\glyphicons-halflings-regular.ttf
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/glyphicons-halflings-regular.eot to fonts\glyphicons-halflings-regular.eot
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/loading.gif to images\loading.gif
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/glyphicons-halflings-regular.woff2 to fonts\glyphicons-halflings-regular.woff2
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/bootstrap.min.css to css\bootstrap.min.css
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/logo.png to images\logo.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/fonts/glyphicons-halflings-regular.svg to fonts\glyphicons-halflings-regular.svg
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/bbd3207c.png to images\bbd3207c.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/icon.png to images\icon.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/star-pattern.png to images\star-pattern.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/background.png to images\background.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/unicorn-map-bg.png to images\unicorn-map-bg.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/css/bootstrap.min.css.map to css\bootstrap.min.css.map
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/unicorn-logo.png to images\unicorn-logo.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/spinning-gears.gif to images\spinning-gears.gif
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-w.png to images\wr-home-w.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-Xiaomi.png to images\wr-home-Xiaomi.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-apple.png to images\wr-home-apple.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-blackberry.png to images\wr-home-blackberry.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-block-1.png to images\wr-home-block-1.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-block-2.png to images\wr-home-block-2.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-about.jpg to images\wr-home-about.jpg
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-block-4.png to images\wr-home-block-4.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-block-3.png to images\wr-home-block-3.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-faq-header.jpg to images\wr-faq-header.jpg
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-google.png to images\wr-home-google.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-facebook.png to images\wr-home-facebook.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-instagram.png to images\wr-home-instagram.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\shadowfox.png to images\shadowfox.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-wechat.png to images\wr-home-wechat.png
download: s3://wildrydes-us-east-1/webApplication/L_StaticwebHosting/website/images\wr-home-weibo.png to images\wr-home-weibo.png
15:37 37
Cloudy ENG IN 30-04-2022
```

```
MINGW64:c/Users/heeha/Desktop/serverless/wildrydes-site
heeha@DESKTOP-9EGL690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ 39°C Cloudy
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images/rocinante.png to images\rocinante.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-top.jpg to images\wr-home-top.jpg
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-note.jpg to images\wr-home-quote.jpg
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-karma.png to images\wr-home-karma.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-2.png to images\wr-investors-2.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-4.png to images\wr-investors-4.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-apply-header.png to images\wr-apply-header.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-1.png to images\wr-investors-1.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-pcp.png to images\wr-investors-pcp.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-3.png to images\wr-investors-3.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-logo-white.png to images\wr-logo-white.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-thebarn.png to images\wr-investors-thebarn.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-logo-black.png to images\wr-logo-black.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-investors-5.png to images\wr-investors-5.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/index.html to \index.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-home-quote.png to images\wr-home-quote.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-unicorn-header.png to images\wr-unicorn-header.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-unicorn-one.png to images\wr-unicorn-one.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/config.js to js\config.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendors.html to js\vendors.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/amazon-cognito-auth.js to js\amazon-cognito-auth.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/main.js to js\main.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/esri-map.js to js\esri-map.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/ride.js to js\ride.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-unicorn-three.png to images\wr-unicorn-three.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/amazon-cognito-identity.min.js to js\vendor\amazon-cognito-identity.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor.js to js\vendor.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/htmlshiv.min.js to js\vendor\htmlshiv.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/bootstrap.min.js to js\vendor\bootstrap.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/modernizr-3.7.1.min.js to js\vendor\modernizr-3.7.1.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/aws-cognito-sdk.min.js to js\vendor\aws-cognito-sdk.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\wr-unicorn-two.png to images\wr-unicorn-two.png
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/respond.min.js to js\vendor\respond.min.js
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/js/vendor/unicorn-icon to js\vendor\unicorn-icon
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/register.html to \register.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/ride.html to \ride.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/messages.txt to messages.txt
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/signin.html to \signin.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/verify.html to \verify.html
download: s3://wildrydes-us-east-1/webapplication/L_StaticwebHosting/website/images\bucephalus.png to images\bucephalus.png
15:37 37
Cloudy ENG IN 30-04-2022
```

Commit the files to your Git repository.

```
MINGW64:c/Users/neeha/Desktop/serverless/wildrydes-site
$ git add .
warning: LF will be replaced by CRLF in apply.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/bootstrap.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/font.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/index.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/main.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/mapbox-gl.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/message.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/normalize.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/ride.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in faq.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in fonts/glyphicons-halfings-regular.svg.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in investors.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/cognito-auth.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/config.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/esri-map.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/amazon-cognito-identity.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/aws-cognito-sdk.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/bootstrap.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/htmlshiv.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/jquery-3.1.0.js.
```

```
MINGW64:c/Users/neeha/Desktop/serverless/wildrydes-site
$ git commit -m "new"
[master (root-commit) 261d03a] new
91 files changed, 13680 insertions(+)
create mode 100644 apply.html
create mode 100644 css/bootstrap.min.css
create mode 100644 css/bootstrap.min.css.map
create mode 100644 css/font.css
create mode 100644 css/index.css
create mode 100644 css/main.css
create mode 100644 css/mapbox-gl.css
create mode 100644 css/message.css
create mode 100644 css/normalize.css
create mode 100644 css/ride.css
create mode 100644 faq.html
create mode 100644 favicon.ico
create mode 100644 fonts/fairplex-wide-n4.woff
create mode 100644 fonts/fairplex-wide-n7.woff
create mode 100644 fonts/glyphicons-halflings-regular.eot
create mode 100644 fonts/glyphicons-halflings-regular.svg
create mode 100644 fonts/glyphicons-halflings-regular.ttf
create mode 100644 fonts/glyphicons-halflings-regular.woff
create mode 100644 fonts/glyphicons-halflings-regular.woff2
create mode 100644 images/background.png
create mode 100644 images/bbd207c.png
create mode 100644 images/bucephalus.png
create mode 100644 images/loading.gif
create mode 100644 images/logo.png
create mode 100644 images/rocinante.png
create mode 100644 images/shadowfox.png
create mode 100644 images/sparkle-gears.gif
create mode 100644 images/star-pattern.png
create mode 100644 images/unicorn-icon.png
create mode 100644 images/unicorn-logo.png
create mode 100644 images/unicorn-map-bg.png
create mode 100644 images/unicorn-silhouette.png
create mode 100644 images/wr-apply-header.png
create mode 100644 images/wr-faq-header.jpg
create mode 100644 images/wr-home-w.png
create mode 100644 images/wr-home-xiaomi.png
create mode 100644 images/wr-home-youtubt.jpg
create mode 100644 images/wr-home-apple.png
create mode 100644 images/wr-home-blackberry.png
create mode 100644 images/wr-home-block-1.png
create mode 100644 images/wr-home-block-2.png
create mode 100644 images/wr-home-block-3.png
create mode 100644 images/wr-home-block-4.png
```

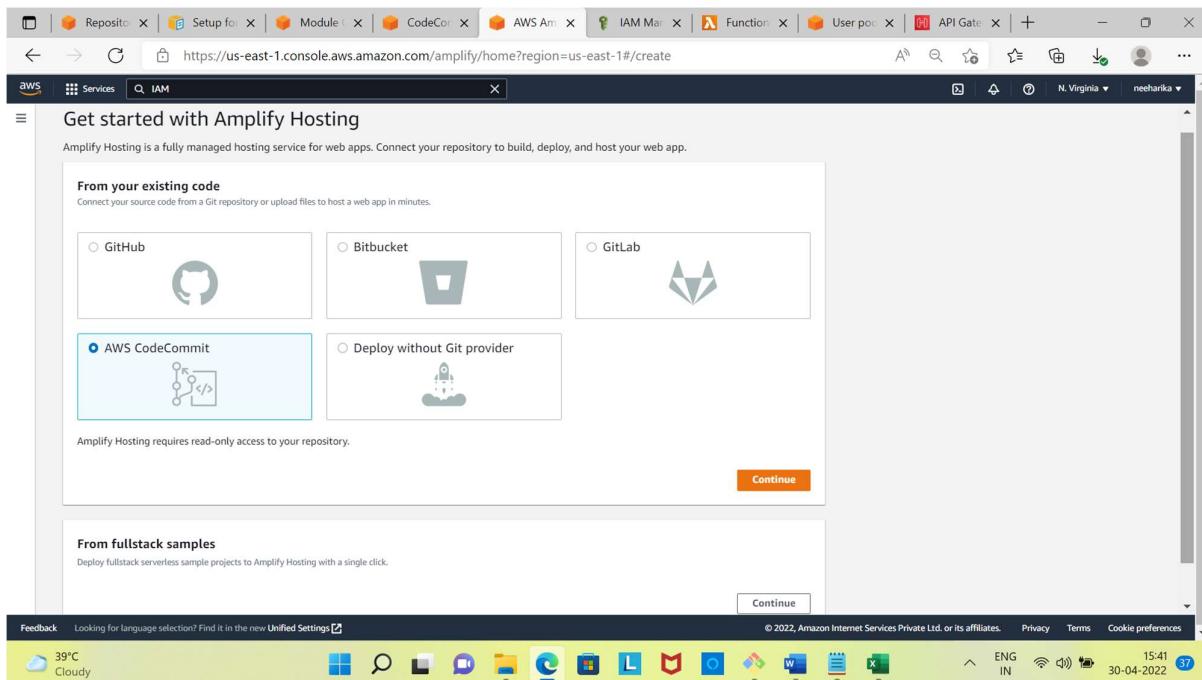
```
MINGW64:/c/Users/neeha/Desktop/serverless/wildrydes-site
$ ls
create mode 100644 images/wr-home-kraken.png
create mode 100644 images/wr-home-quote.jpg
create mode 100644 images/wr-home-quote.png
create mode 100644 images/wr-home-top.jpg
create mode 100644 images/wr-home-twitter.png
create mode 100644 images/wr-home-wechat.png
create mode 100644 images/wr-home-weibo.png
create mode 100644 images/wr-investors-1.png
create mode 100644 images/wr-investors-2.png
create mode 100644 images/wr-investors-3.png
create mode 100644 images/wr-investors-4.png
create mode 100644 images/wr-investors-5.png
create mode 100644 images/wr-investors-awesome.png
create mode 100644 images/wr-investors-header.png
create mode 100644 images/wr-investors-pcp.png
create mode 100644 images/wr-logo-black-transparent.png
create mode 100644 images/wr-investors-2.png
create mode 100644 images/wr-investors-3.png
create mode 100644 images/wr-investors-4.png
create mode 100644 images/wr-investors-5.png
create mode 100644 index.html
create mode 100644 investors.html
create mode 100644 js/cognito-auth.js
create mode 100644 js/config.js
create mode 100644 js/esri-map.js
create mode 100644 js/main.js
create mode 100644 js/ride.js
create mode 100644 js/vendor.js
create mode 100644 js/vendor/amazon-cognito-identity.min.js
create mode 100644 js/vendor/aws-cognito-sdk.min.js
create mode 100644 js/vendor/bootstrap.min.js
create mode 100644 js/vendor/html5shiv.min.js
create mode 100644 js/vendor/html5shiv-test.js
create mode 100644 js/vendor/modernizr.js
create mode 100644 js/vendor/moment.min.js
create mode 100644 js/vendor/respond.min.js
create mode 100644 js/vendor/unicorn-icon'
create mode 100644 register.html
create mode 100644 ride.html
create mode 100644 robots.txt
create mode 100644 signin.html
create mode 100644 unicorns.html
create mode 100644 verify.html
neeha@DESKTOP-9EGL690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ |
```

Cloudy 39°C ENG IN 15:39 37 30-04-2022

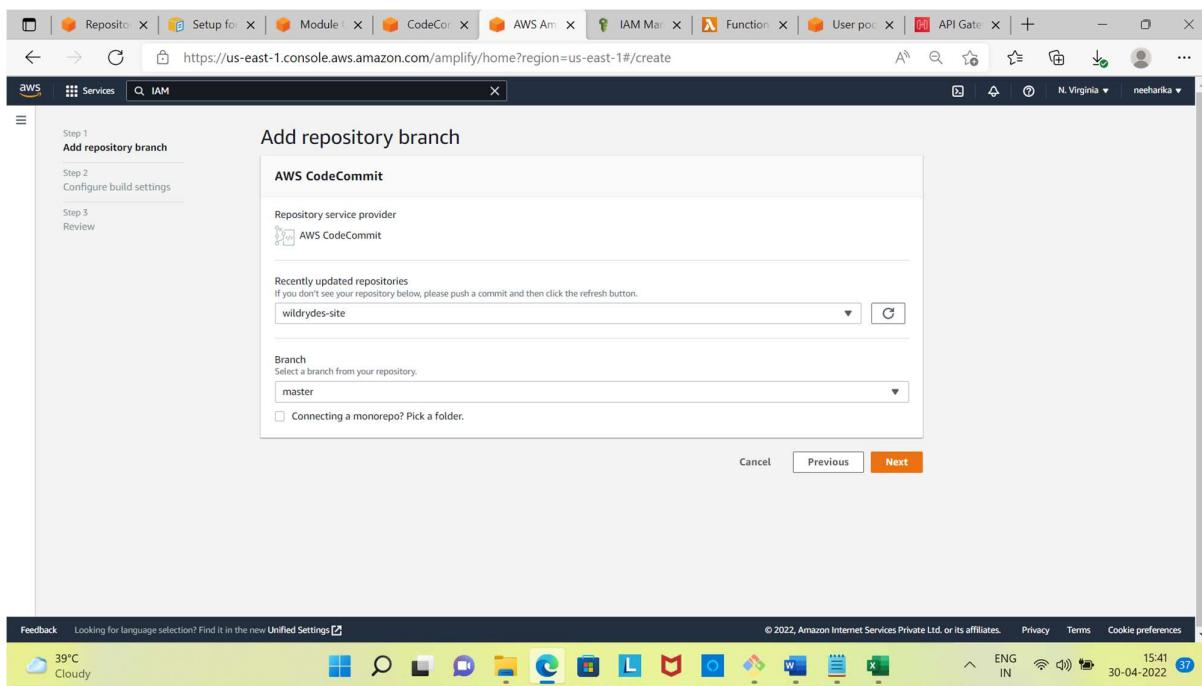
```
MINGW64:/c/Users/neeha/Desktop/serverless/wildrydes-site
$ git push
Enumerating objects: 95, done.
Counting objects: 100% (95/95), done.
Delta compression using up to 8 threads
Compressing objects: 100% (95/95), done.
Writing objects: 100% (95/95), 9.44 MiB | 3.19 MiB/s, done.
Total 95 (delta 2), reused 0 (delta 0), pack-reused 0
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
 * [new branch] master -> master
neeha@DESKTOP-9EGL690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$
```

Cloudy 39°C ENG IN 15:39 37 30-04-2022

3. Enable Web hosting with AWS Amplify Console – Launch AWS Amplify and select the code commit repository created.



On the "Configure build settings" page leave all the defaults and select Next.



On the "Review" page select Save and deploy. Process takes a couple of minutes for Amplify Console to create the necessary resources and to deploy your code.

Step 1
Add repository branch

Step 2
Configure build settings

Step 3
Review

Review

Repository details

Repository service: AWS CodeCommit

Repository: wildrydes-site

Branch: master

Branch environment: Application root

App settings

App name: wildrydes-site

Framework: Web

Build image: Using default image

Build settings: Auto-detected settings will be used

Environment variables: None

Cancel Previous Save and deploy

Once completed, click on the site image to launch your Wild Rydes site.

All apps > wildrydes-site

wildrydes-site

This tab lists all connected branches, select a branch to view build details.

master

Continuous deploy set up (Edit)

https://master...amplifyapp.com

Provision Build Deploy Verify

Last deployment: 4/30/2022, 4:10:12 PM

Last commit: This is an autogenerated message | Auto-build | AWS CodeCommit - master

Connect branch



Modify your site - Open the `index.html` page and modify the title line so that it says: <title>Wild Rydes - Rydes of the Future!</title>.

Save the file and commit to your git repository again.

```
neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git commit -m "updated title"
[master 77ae41c] updated title
 1 file changed, 1 insertion(+), 1 deletion(-)

neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
 261d03a..77ae41c master -> master

neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
```

Re-open the Wild Rydes site and notice the title change.

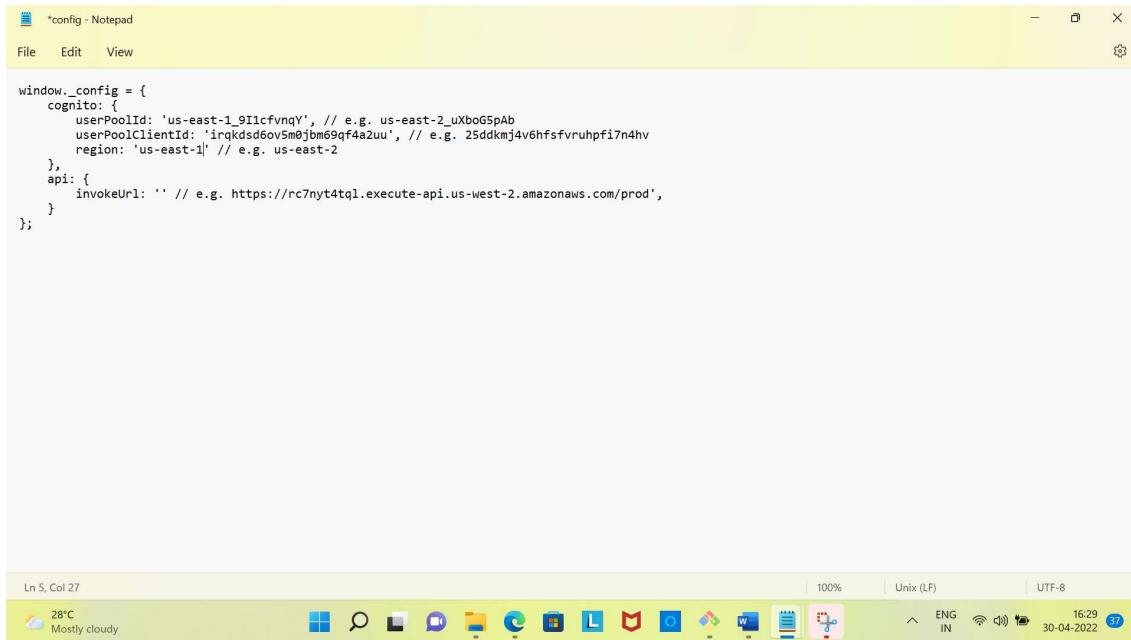


Step2: User Management

1. Create an Amazon Cognito user pool and add an App client to it.

User pool name	User pool ID	Created time	Last updated time
WildRydes	us-east-1_9l1cfvnqY	13 seconds ago	13 seconds ago

2. Update website Config. From your local machine, open `wild-ryde-site/js/config.js` in a text editor of your choice and update Cognito section with correct values for the userpool and app you created.

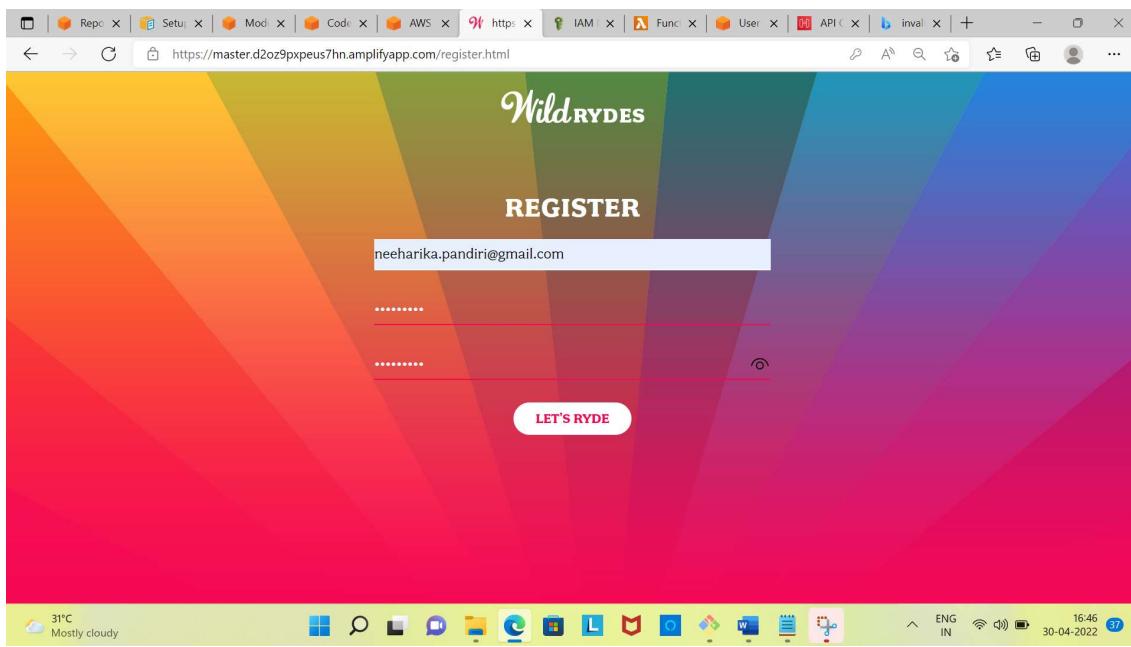


```
window._config = {
  cognito: {
    userPoolId: 'us-east-1_9I1cfvnqY', // e.g. us-east-2_uXboG5pAb
    userPoolClientId: 'irqkdsd6ov5m0jbm69qf4a2uu', // e.g. 25ddkmj4v6hfsfvruhpfi7n4hv
    region: 'us-east-1' // e.g. us-east-2
  },
  api: {
    invokeUrl: '' // e.g. https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod',
  }
};
```

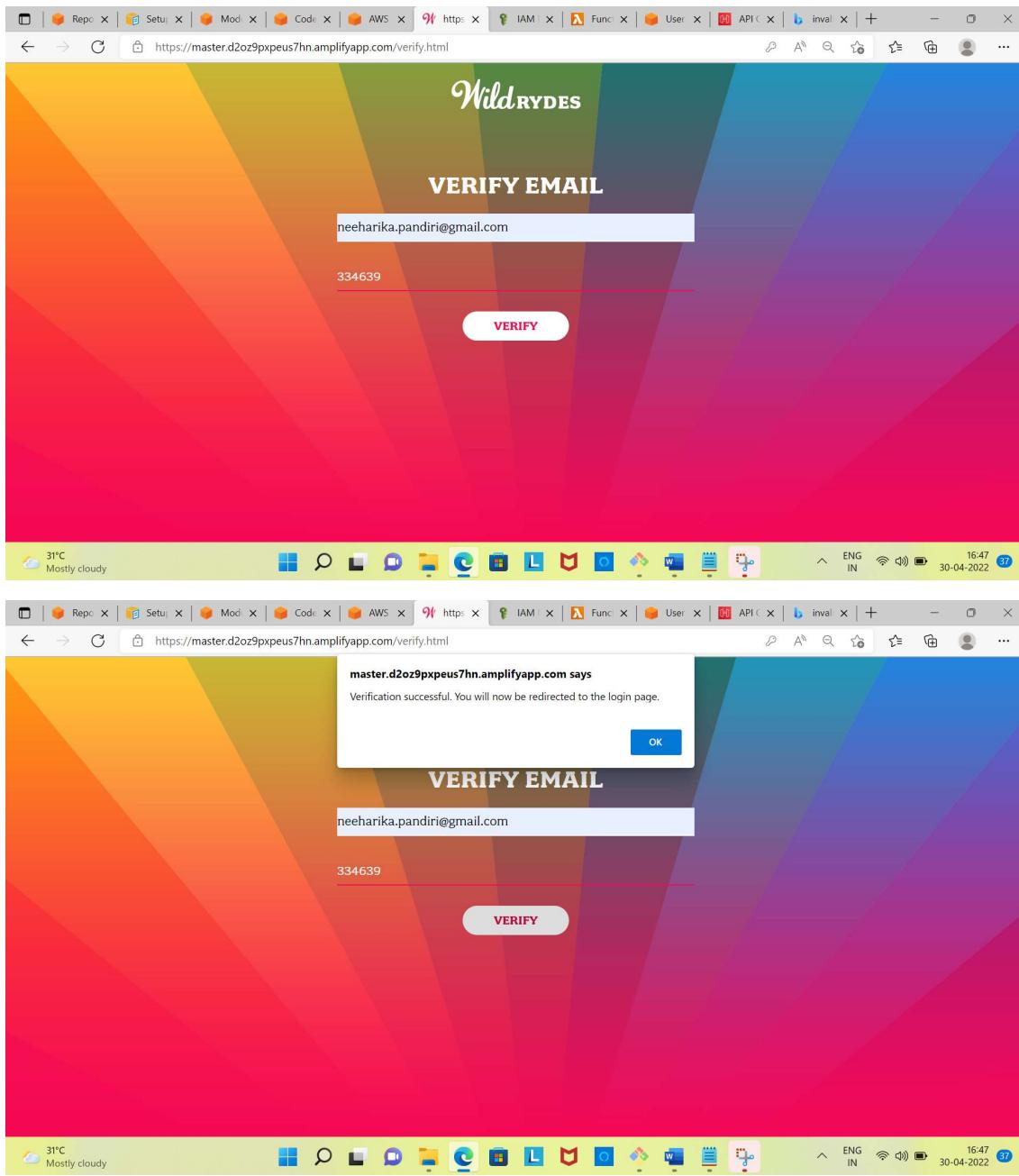
Save the modified file and push it to your Git repository to have it automatically deploy to Amplify Console.

3. Validate the implementation for the changes you made.

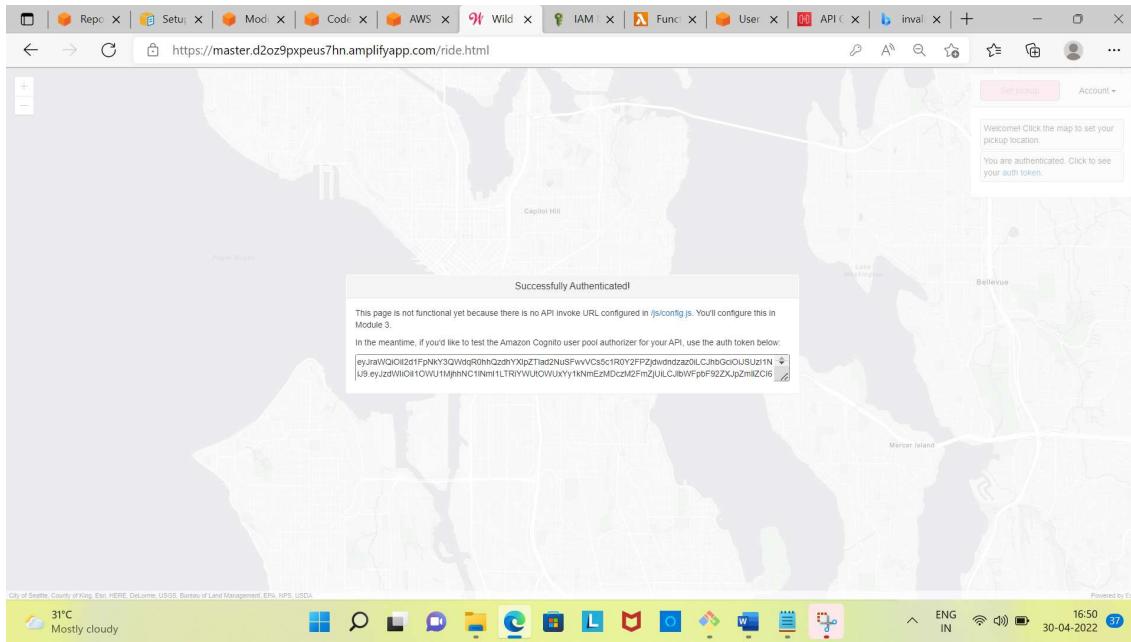
Visit </register.html> and complete registration form



Complete the account verification process by visiting </verify.html>



Visit `/signin.html` and log in using the email address and password you entered during the registration step. After successful authentication, you are redirected to `/ride.html` page.



Step3: Serverless Backend

1. Create an Amazon DynamoDB table with name 'Rides' and Partition key 'RideId'.

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Size	Table class
Rides	Active	RideId (S)	-	0	Provisioned with auto scaling (5)	Provisioned with auto scaling (5)	0 bytes	DynamoDB Standard

The screenshot shows the AWS DynamoDB console with the Rides table selected. The left sidebar includes options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Reserved capacity, DAX Clusters, Subnet groups, Parameter groups, Events, Tell us what you think, Return to the previous console experience, and Density settings. The main panel displays the Rides table's General information, including Partition key (RideId String), Sort key (-), Capacity mode (Provisioned), and Table status (Active, No active alarms). It also shows Additional info such as Table class (DynamoDB Standard), Indexes (0 global, 0 local), DynamoDB stream (Disabled), Point-in-time recovery (PITR) (Disabled), Time to Live (TTL) (Disabled), Replication Regions (0 Regions), Encryption (Owned by Amazon), Date created (April 30, 2022, 16:58:46 UTC+05:30), and Amazon Resource Name (ARN) (arn:aws:dynamodb:us-east-1:689280071887:table/Rides).

2. Create an IAM role for your Lambda Function. Attach the managed policy called *AWSLambdaBasicExecutionRole* to this role to grant the necessary CloudWatch Logs permissions. Also, create a custom inline policy for your role that allows the *ddb:PutItem* action for the table you created.

The screenshot shows the AWS IAM console with the WildRydesLambda role selected. The left sidebar includes Identity and Access Management (IAM) options like Dashboard, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, Analyzes, Settings, Credential report, Organization activity, and Service control policies (SCPs). The main panel displays the WildRydesLambda role's Summary, showing Creation date (April 30, 2022, 17:06 UTC+05:30), Last activity (None), ARN (arn:aws:iam:689280071887:role/WildRydesLambda), and Maximum session duration (1 hour). The Permissions tab lists attached policies: AmazonAPIGatewayPushToCloudWatchLogs, CloudWatchFullAccess, AWSLambdaBasicExecutionRole, and a custom inline policy named ddbPutItem. The Permissions boundary section is noted as '(not set)'.

3. Create a new Lambda function called *RequestUnicorn* that will process the API requests.

Create function

Choose one of the following options to create your function.

- Author from scratch** (selected): Start with a simple Hello World example.
- Use a blueprint**: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image**: Select a container image to deploy for your function.
- Browse serverless app repository**: Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Runtime
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Execution role
Choose an IAM role that defines the permissions Lambda needs to run your function. To learn more about roles, go to the IAM console.

Use an existing role
 Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

Advanced settings

Enable Code signing

Enable function URL - new

Auth type
Choose the auth type for your function URL. Learn more [\[?\]](#)

AWS_IAM
Only authenticated IAM users and roles can make requests to your function URL.

NONE
Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

Function URL permissions

When you choose auth type **NONE**, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type **AWS_IAM**.

View policy statement

Configure cross-origin resource sharing (CORS)

Enable VPC

Feedback: Looking for language selection? Find it in the new Unified Settings [\[?\]](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

ENG IN 17:18 30-04-2022

4. Validate the function using Test event.

Screenshot of the AWS Lambda console showing the successful update of the function RequestUnicorn.

Test event

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

- Create new event
- Edit saved event

Event name

TestRequestEvent

Event JSON

```

1 [ {
2   "path": "/ride",
3   "httpMethod": "POST",
4   "headers": {
5     "Accept": "*/*",
6     "Authorization": "eyJraeiQIoi3LTzRW%Zs",
7     "Content-Type": "application/json; charset=UTF-8"
8   },
9   "queryStringParameters": null,
10  "pathParameters": null,
11  "requestContext": {
12    "authorizer": {
13      "claims": {
14        "cognito:username": "the_username"
15      }
16    }
17  },
18  "body": "{\"pickupLocation\":{\"Latitude\":47.6174755835663,\"Longitude\":-122.28837066650185}}"
19 }]

```

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 17:23 30-04-2022

Execution result: succeeded (logs)

The area below shows the last 4 KB of the execution log.

```

{
  "statusCode": 200,
  "body": "{\"name\":\"Unicorn\", \"color\":\"yellow\", \"gender\":\"female\", \"UnicornName\":\"Kocinante\", \"Eta\": \"38 seconds\", \"Ride\":\"The Unicorn\"}",
  "headers": {
    "Access-Control-Allow-Origin": "*"
  }
}

Summary
Code SHA-256
pw79PfX00EiNAPq4h+mGark8RBD0mwfhdYSCwU4-
Init duration
460.86 ms
Billed duration
981 ms
Max memory used
78 MB
Log output
The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.
START RequestId: af73acd-029-4008-a2b-6002f5d520dc Version: $LATEST
2022-04-30T15:09:09,000 af73acd-029-4008-a2b-6002f5d520dc INFO Received event ( $lAMBDA$stage$prefix$event ): {
  path: '/ride',
  httpMethod: 'POST',
  headers: {
    accept: '*/*',
    authorization: 'eyJraeiQIoi3LTzRW%Zs',
    content-type: 'application/json; charset=UTF-8'
  },
  queryStringParameters: null,
}

```

Test event

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences ENG IN 17:24 30-04-2022

Step 4: Deploy REST API

1. Create new REST API using Amazon API Gateway.

Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

REST WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints

New API Import from Swagger or Open API 3 Example API

Settings

Choose a friendly name and description for your API.

API name* WildRydes

Description

Endpoint Type Edge optimized

* Required

Create API



2. Create Cognito user pool authorizer and test the configuration using auth token you received on /signin.html.

WildRydes - Test Authorizer

Response Code: 200

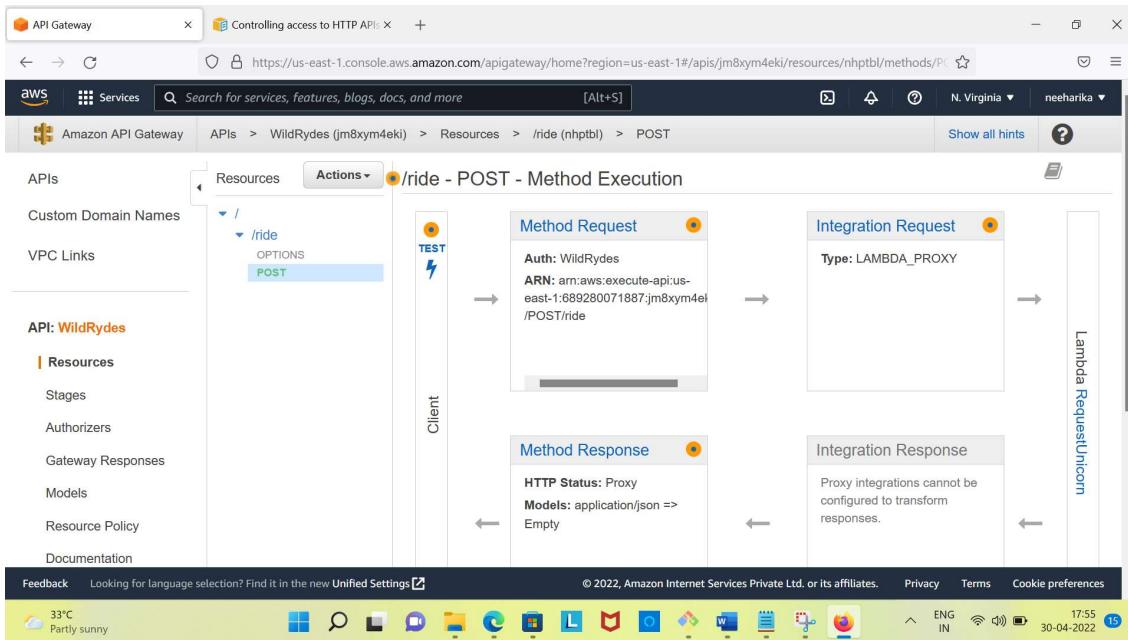
Latency 53

Claims

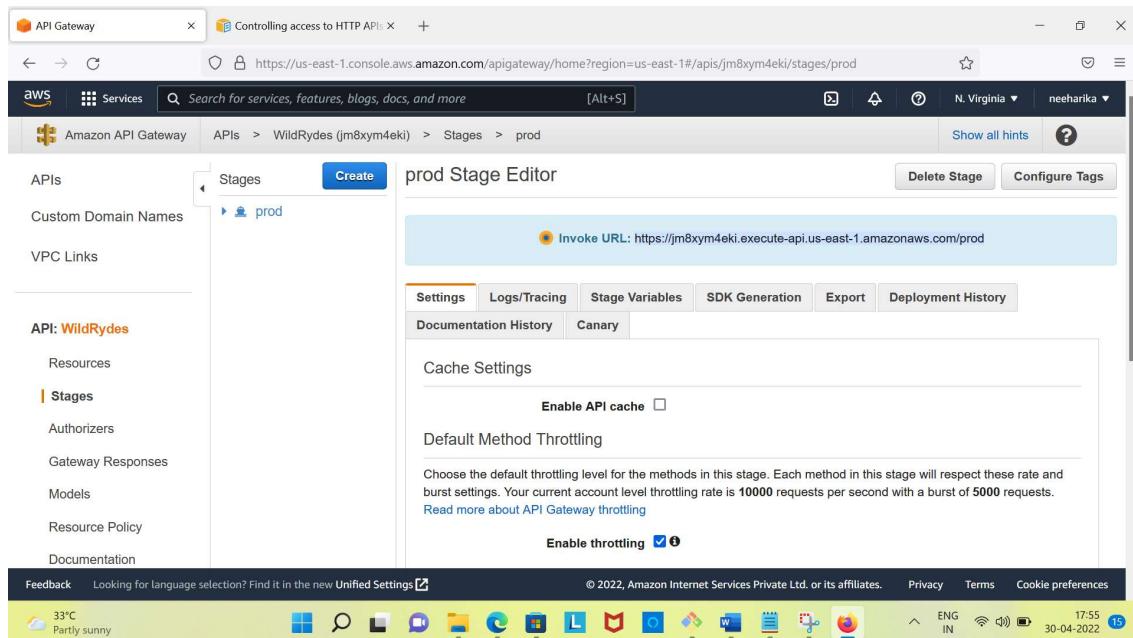
```
{
  "aud": "irqkdsd6ov5m0jbm69qf4a2uu",
  "auth_time": "1651317596",
  "cognito:username": "59e528a4-e6b5-4bae-9e1c-d6a30733aff5",
  "email": "neeharika.pandiri@gmail.com",
  "email_verified": "true",
  "event_id": "72282bd9-abc1-49d7-b652-93393e3e66a3",
  "exp": "Sat Apr 30 12:19:56 UTC 2022",
  "iat": "Sat Apr 30 11:19:56 UTC 2022",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_9I1cfvNqY",
  "jti": "d0718f4d-472d-4488-b209-ce3ac566cc30",
  "origin_jti": "3c3fe220-435a-4e13-819d-c914d381c529",
  "sub": "59e528a4-e6b5-4bae-9e1c-d6a30733aff5",
  "token_use": "id"
}
```

Close

3. Create a new resource called /ride within your API. Then create a POST method for that resource and configure it to use a Lambda proxy integration backed by the RequestUnicorn function you created.



4. Deploy your API and create a new stage 'prod'. Note the invoke URL.



5. Update the invokeUrl setting under the api key in the config.js file. Set the value to the Invoke URL for the deployment stage you created in previous step.



```
config - Notepad
File Edit View
window._config = {
  cognito: {
    userPoolId: 'us-east-1_9I1cfvnqY', // e.g. us-east-2_uXboG5pAb
    userPoolClientId: 'irqkdsd6ovSm0jbm69qf4a2uu', // e.g. 25ddkmj4v6hfsfvruhpfi7n4hv
    region: 'us-east-1' // e.g. us-east-2
  },
  api: {
    invokeUrl: 'https://jm8xym4eki.execute-api.us-east-1.amazonaws.com/prod' // e.g. https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod',
  }
};
```

Save the modified file and push it to your Git repository to have it automatically deploy to Amplify Console.

```
neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git add .
warning: LF will be replaced by CRLF in js/config.js.
The file will have its original line endings in your working directory

neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git commit -m "config API invoke URL"
[master 500455c] config API invoke URL
 1 file changed, 1 insertion(+), 1 deletion(-)

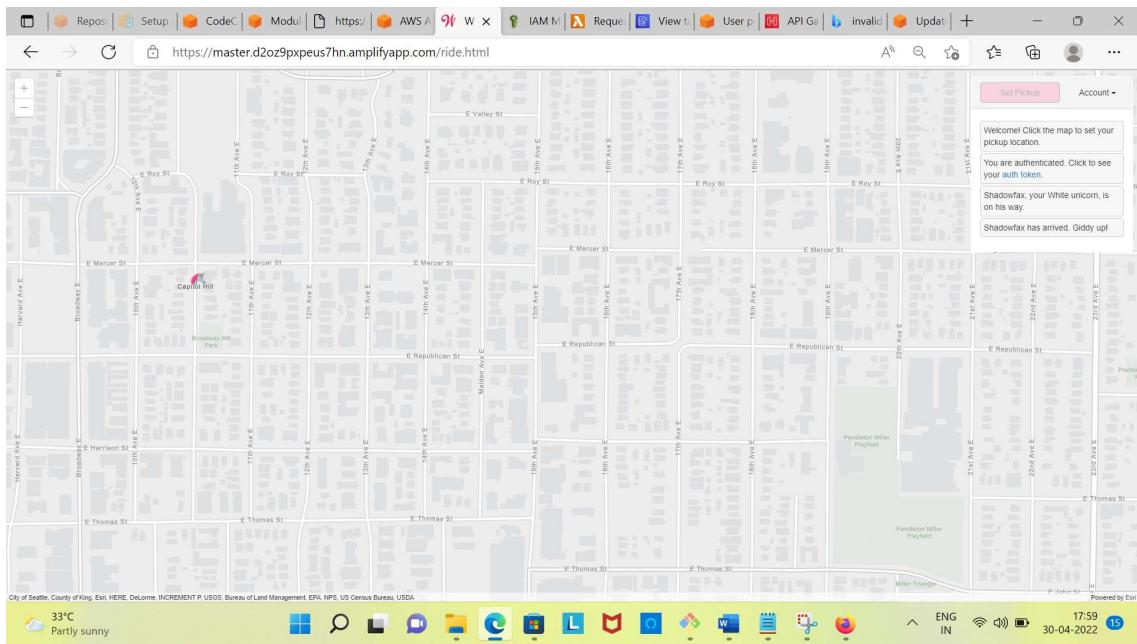
neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 426 bytes | 426.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
 27d1b91..500455c master -> master

neeha@DESKTOP-9E6L690 MINGW64 ~/Desktop/serverless/wildrydes-site (master)
$ |
```

Validate Implementation

Visit */ride.html* under your website domain.

- b. If you are redirected to the sign in page, sign in with the user you created.
- c. After the map has loaded, click anywhere on the map to set a pickup location.
- d. Choose Request Unicorn. You should see a notification in the right sidebar that a unicorn is on its way and then see a unicorn icon fly to your pickup location.



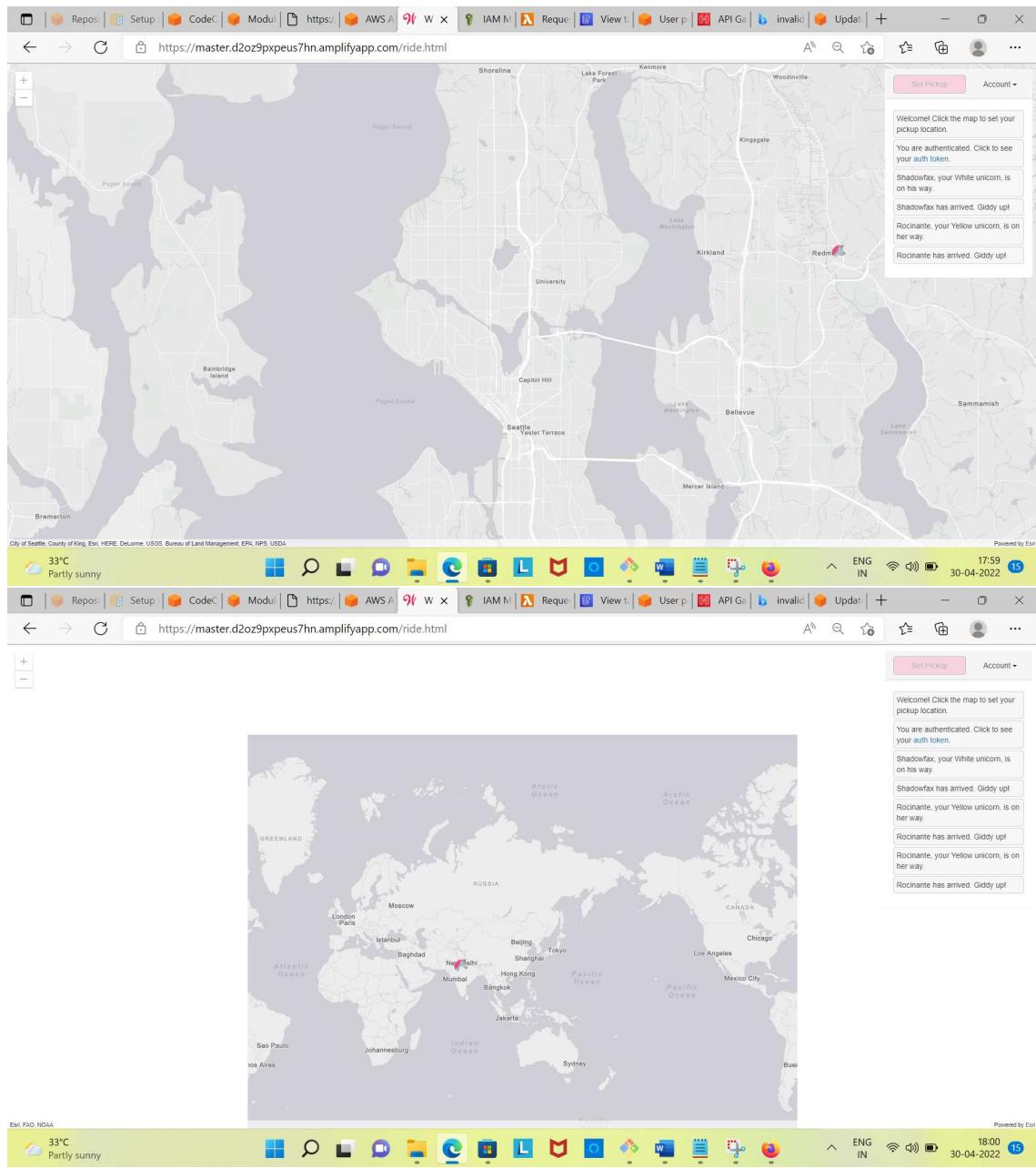
City of Seattle, County of King, Est. HERE, DeLorme, INCREMENT P, USGS, Bureau of Land Management, EPA, NPS, US Census Bureau, USDA.

33°C
Partly sunny



17:59
30-04-2022





PFB details of rides registered in Dynamo DB table.

DynamoDB

Scan/Query items

Scan Query

Table or index Rides

Filters

Run Reset

Completed Read capacity units consumed: 0.5

Items returned (4)

Rideld	RequestTime	Unicorn	Unicorn...	User
HSQuYZW658KGr4M6HDg	2022-04-30T12:29:49.487Z	{"Name": {"S": "Rocinante"}, "Color": {}}	Rocinante	59e52ba4-e6b5-4b..
z5SybXf03b8A8nbywlvYmQ	2022-04-30T12:30:50.126Z	{"Name": {"S": "Rocinante"}, "Color": {}}	Rocinante	59e52ba4-e6b5-4b..
OjK3B6GwYg5RIt3Ht5DvEA	2022-04-30T11:53:05.055Z	{"Name": {"S": "Rocinante"}, "Color": {}}	Rocinante	the_username
M_AzNqzEvGIGJ0FT8rPvRoQ	2022-04-30T12:28:58.645Z	{"Name": {"S": "Shadowfax"}, "Color": {}}	Shadowfax	59e52ba4-e6b5-4b..

Tell us what you think
Return to the previous console experience
Density settings

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

ENG IN 18:02 30-04-2022

Cost Analysis

AWS amplify	Build & Deploy Charges	Total build time per month = num of commits/day * num of days * avg. build time = $2*20*3 = 120$ build mins per month
		Monthly build & deploy charges = $120 * .01\text{USD} = 1.2 \text{ USD}$
	Hosting Charges	Web app Size= 25MB, Avg Size of Page Requested = 1MB
		Data Stored = Web app Size * num of monthly builds = $(25/1024) * (2*20) = 0.97\text{GB} \sim 1\text{GB}$
		Data Served = Daily active users * average page size * days = $1000*(1/1024)*30 = 29.29 \text{ GB} \sim 30\text{GB}$
		Data Stored cost = $1*0.023\text{USD} = 0.023 \text{ USD}$
		Data served cost = $30*0.15\text{USD} = 4.5 \text{ USD}$

Cognito User pool	1,000 MAUs x 0.10 SAML or OIDC federation requests = 100.00 SAML or OIDC federation MAU requests
	100.00 SAML or OIDC federation MAUs - 50 free SAML or OIDC federation MAU requests per month = 50.00 billable SAML or OIDC federation MAU requests
	Max (50.00 billable SAML or OIDC federation MAU requests, 0 minimum billable SAML or OIDC federation MAU requests) = 50 total billable SAML or OIDC federation MAU requests
	50 MAUs x 0.015 USD = 0.75 USD (SAML or OIDC federation MAU requests)
	SAML or OIDC federation cost (monthly): 0.75 USD
	1,000 MAUs - 50000 free MAU requests per month = -49,000.00 billable MAU requests

	Tiered price for: -49000.000000 MAUs
	Total tier cost = 0.00 USD (User Pool MAUs)
	User Pool MAU cost (monthly): 0.00 USD
	Advanced security feature cost (monthly): 0 USD
Cognito MAU cost (monthly): 0.75 USD	

Lambda	Number of requests: 1000 per hour * (730 hours in a month) = 730000 per month
	Amount of memory allocated: 128 MB x 0.0009765625 GB in a MB = 0.125 GB
	Amount of ephemeral storage allocated: 512 MB x 0.0009765625 GB in a MB = 0.5 GB
	<u>Pricing calculations</u>
	730,000 requests x 1,000 ms x 0.001 ms to sec conversion factor = 730,000.00 total compute (seconds)
	0.125 GB x 730,000.00 seconds = 91,250.00 total compute (GB-s)
	91,250.00 GB-s x 0.0000166667 USD = 1.52 USD (monthly compute charges)
	730,000 requests x 0.0000002 USD = 0.15 USD (monthly request charges)
	0.50 GB - 0.5 GB (no additional charge) = 0.00 GB billable ephemeral storage per function
	1.52 USD + 0.15 USD = 1.67 USD
Lambda costs - Without Free Tier (monthly): 1.67 USD	

REST API	1 requests x 1,000,000 unit multiplier = 1,000,000 total REST API requests
	Tiered price for: 1000000 requests
	1000000 requests x 0.0000035000 USD = 3.50 USD
	Total tier cost = 3.5000 USD (REST API requests)
	Tiered price total for REST API requests: 3.50 USD
	0 USD per hour x 730 hours in a month = 0.00 USD for cache memory
	Dedicated cache memory total price: 0.00 USD
REST API cost (monthly): 3.50 USD	

DynamoDB - Provisioned Capacity - 5RCU & 5WCU	Price per hour	Price for Provisioned Capacity Per hour	Price for Provisioned Capacity per month
	\$0.00065 per RCU	0.00325	2.34
	\$0.00013 per RCU	0.00065	0.468
Dynamo DB Provisioned cost (monthly): 2.8 USD			

Total Solution cost = 15 USD