CS55A

Jason Palmeri

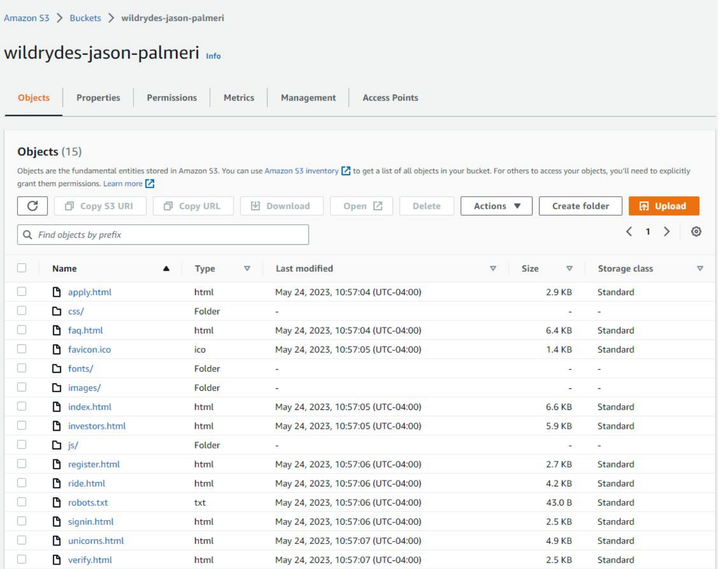
Building the Platform - AWS S3

Amazon Simple Storage Service (S3) is a scalable, durable, and highly available storage service that allows us to store files on the cloud, and enable static website hosting.

S3 provides us with all the tools we need to upload, download and configure our web hosting to get WildRydes up and running within minutes.

With S3's scalability and durability we should expect smooth transitions to larger data sets and more users, while keeping costs low on a pay-as-you-go model

http://wildrydes-jason-palmeri.s3-website-us-east-1.amazonaws.com/



wildrydes-jason-palmeri Info

Objects (15)

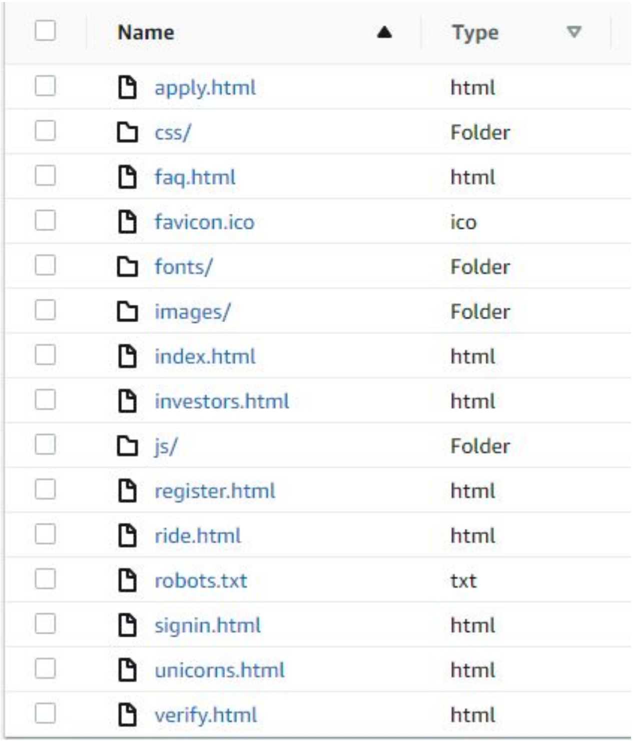
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type	Last modified	Size	Storage class
□	apply.html	html	May 24, 2023, 10:57:04 (UTC-04:00)	2.9 KB	Standard
□	css/	Folder	-	-	-
□	faq.html	html	May 24, 2023, 10:57:04 (UTC-04:00)	6.4 KB	Standard
□	favicon.ico	ico	May 24, 2023, 10:57:05 (UTC-04:00)	1.4 KB	Standard
□	fonts/	Folder	-	-	-
□	images/	Folder	-	-	-
□	index.html	html	May 24, 2023, 10:57:05 (UTC-04:00)	6.6 KB	Standard
□	investors.html	html	May 24, 2023, 10:57:05 (UTC-04:00)	5.9 KB	Standard
□	js/	Folder	-	-	-
□	register.html	html	May 24, 2023, 10:57:06 (UTC-04:00)	2.7 KB	Standard
□	ride.html	html	May 24, 2023, 10:57:06 (UTC-04:00)	4.2 KB	Standard
□	robots.txt	txt	May 24, 2023, 10:57:06 (UTC-04:00)	43.0 B	Standard
□	signin.html	html	May 24, 2023, 10:57:06 (UTC-04:00)	2.5 KB	Standard
□	unicorns.html	html	May 24, 2023, 10:57:07 (UTC-04:00)	4.9 KB	Standard
□	verify.html	html	May 24, 2023, 10:57:07 (UTC-04:00)	2.5 KB	Standard

AWS S3 - In-Depth

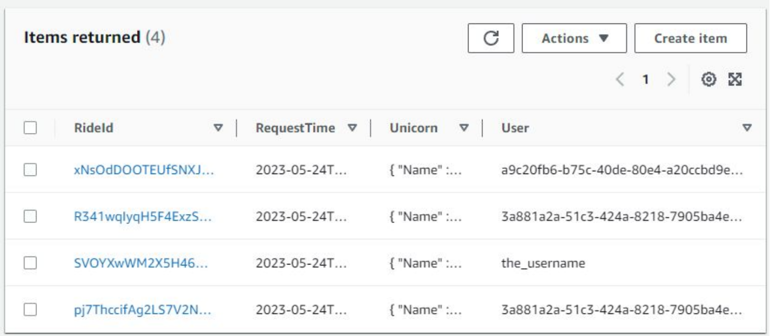
AWS S3 is one of many ways to host and deploy a website or web application. For WildRydes we use S3 as our core website is a static website, meaning there isn’t a whole lot of change going on in the website regularly.

AWS S3 hosts our web applications file on the cloud, this includes all images, fonts, layout styles, and web pages. Having our files hosted on the cloud allows our developers to easily access code from all around the world.



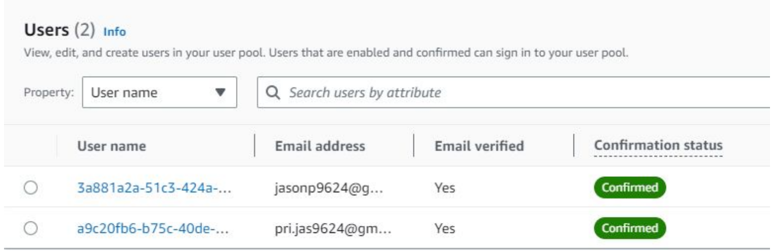
□	Name ▲	Type ▽
□	apply.html	html
□	css/	Folder
□	faq.html	html
□	favicon.ico	ico
□	fonts/	Folder
□	images/	Folder
□	index.html	html
□	investors.html	html
□	js/	Folder
□	register.html	html
□	ride.html	html
□	robots.txt	txt
□	signin.html	html
□	unicorns.html	html
□	verify.html	html

Building the Platform - DynamoDB and Cognito Users



Items returned (4)

☐	RideId	RequestTime	Unicorn	User
☐	xNsOdDOOTEUfSNXJ...	2023-05-24T...	{ "Name" :...	a9c20fb6-b75c-40de-80e4-a20ccbd9e...
☐	R341wqlyqH5F4ExzS...	2023-05-24T...	{ "Name" :...	3a881a2a-51c3-424a-8218-7905ba4e...
☐	SVOYXwWM2X5H46...	2023-05-24T...	{ "Name" :...	the_username
☐	pj7ThccifAg2LS7V2N...	2023-05-24T...	{ "Name" :...	3a881a2a-51c3-424a-8218-7905ba4e...



Users (2) Info

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Property: User name ▼ Search users by attribute

	User name	Email address	Email verified	Confirmation status
○	3a881a2a-51c3-424a-...	jasonp9624@g...	Yes	Confirmed
○	a9c20fb6-b75c-40de-...	pri.jas9624@gm...	Yes	Confirmed

DynamoDB is one of AWS's database services that allows us to store Ride data for future analysis and support for users. Within our Rides database we store all user transactions, detailing the request time, unicorn information and the user that requested the ride.

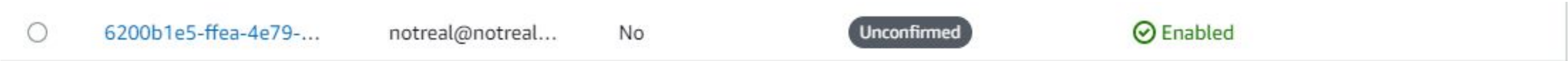
User data stored in the Ride database comes from AWS's Cognito User Pool, which allows us to setup user registration and sign-in seamlessly into the WildRydes application, with built in confirmation and email services.

Together these services allow us to track users who request rides, as well as allow new users to register.

AWS Cognito In-Depth

AWS Cognito is our all in one user management system, and handles all of our needs for the time being. Cognito is built into our application using their JavaScript libraries to provide user registration, user logins, and user authentication.

With AWS Cognito we can see a list of our users, and their statuses, only allowing logged in, and confirmed users access the application.

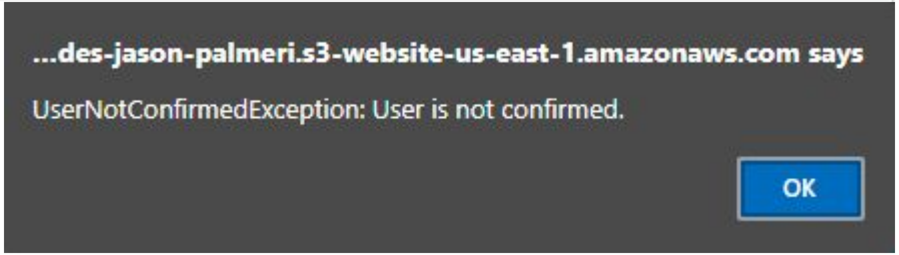


○	6200b1e5-ffea-4e79-...	notreal@notreal...	No	Unconfirmed	✓ Enabled

Cognito will help us to support our customers

Who are having issues by allowing our team to see

Their current status and other user information

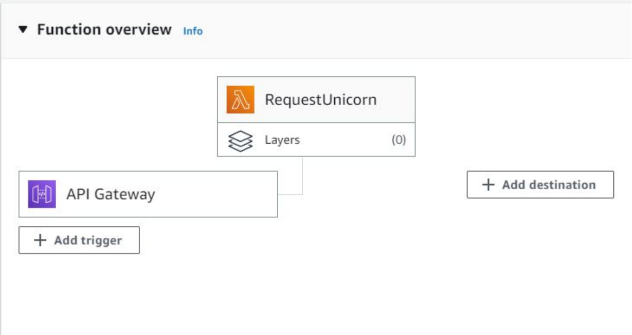


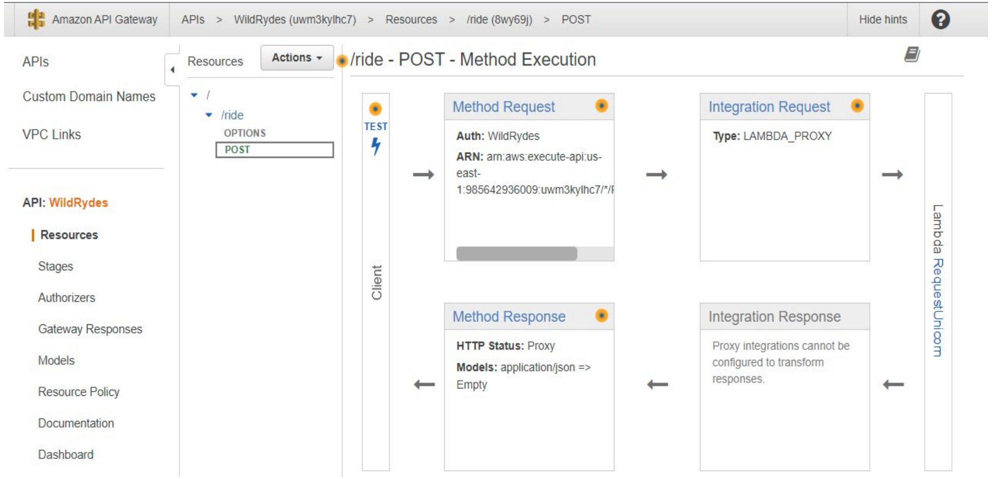
...des-jason-palmeri.s3-website-us-east-1.amazonaws.com says

UserNotConfirmedException: User is not confirmed.

Building the Platform - Lambda and the API Gateway

AWS Lambda is a service that allows us to run code based on certain actions, or triggers that occur. With the AWS API Gateway we can send ride requests from the app to the Lambda functions, which will generate ride data for our users.





/ride - POST - Method Execution

Method Request

Auth: WildRydes

ARN: arn:aws:execute-api:us-east-1:985642936009:uwm3kylhc7/*/P

Integration Request

Type: LAMBDA_PROXY

Method Response

HTTP Status: Proxy

Models: application/json => Empty

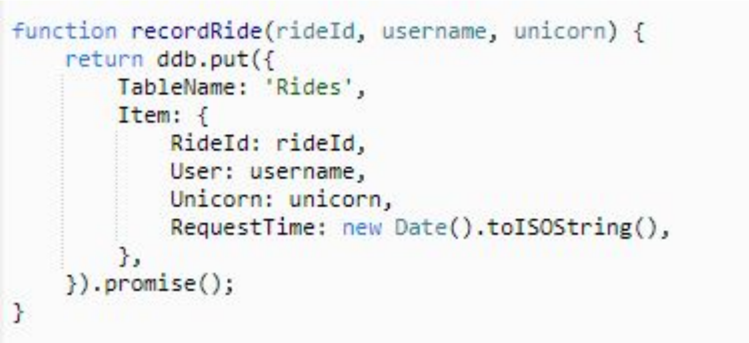
Integration Response

Proxy integrations cannot be configured to transform responses.

AWS Lambda - In-Depth

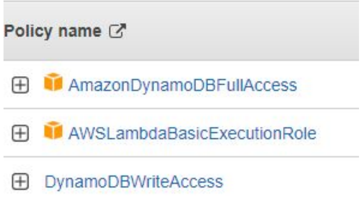
AWS Lambda is an important platform for WildRydes as it handles our database writing, and also sends data to our web application based on API triggers.

Currently our Lambda platform holds the RequestUnicorn function, which when requested by our API, writes data to our database, and sends the user information on their requested ride.



```
function recordRide(rideId, username, unicorn) {
    return ddb.put({
        TableName: 'Rides',
        Item: {
            RideId: rideId,
            User: username,
            Unicorn: unicorn,
            RequestTime: new Date().toISOString(),
        },
    }).promise();
}
```

Within AWS Lambda we use a custom IAM role for our Lambda: WildRydesLambda. This role allows us to access other AWS services such as DynamoDB giving us full access to create, read, update, and delete entries as needed.

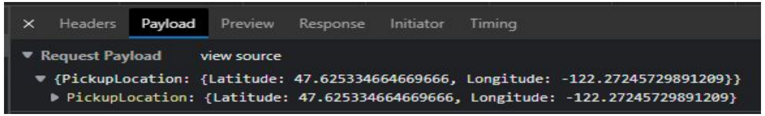


Policy name
⊞ AmazonDynamoDBFullAccess
⊞ AWSLambdaBasicExecutionRole
⊞ DynamoDBWriteAccess

AWS API Gateway - In-Depth

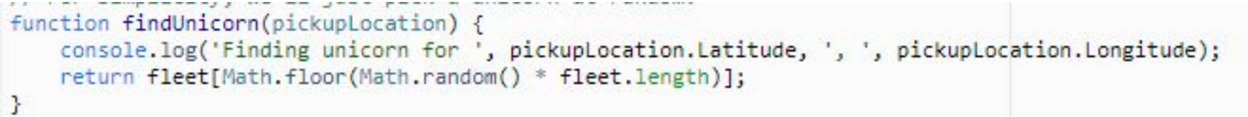
The AWS API Gateway allows us to make calls to other services and functions to do certain tasks. In WildRydes we use the /ride route to make a request to our lambda function. This goes through a few steps to get the requested data:

1. Make a POST request to /ride with PickupLocation



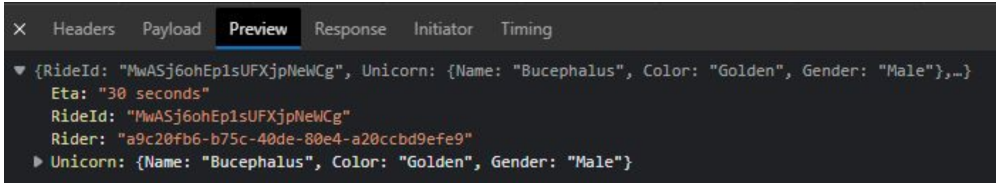
```
▼ Request Payload      view source
  ▼ {PickupLocation: {Latitude: 47.625334664669666, Longitude: -122.27245729891209}}
    ▶ PickupLocation: {Latitude: 47.625334664669666, Longitude: -122.27245729891209}
```

2. The POST request then sends this data to our findUnicorn function within our Lambda to find a unicorn



```
function findUnicorn(pickupLocation) {
    console.log('Finding unicorn for ', pickupLocation.Latitude, ', ', pickupLocation.Longitude);
    return fleet[Math.floor(Math.random() * fleet.length)];
}
```

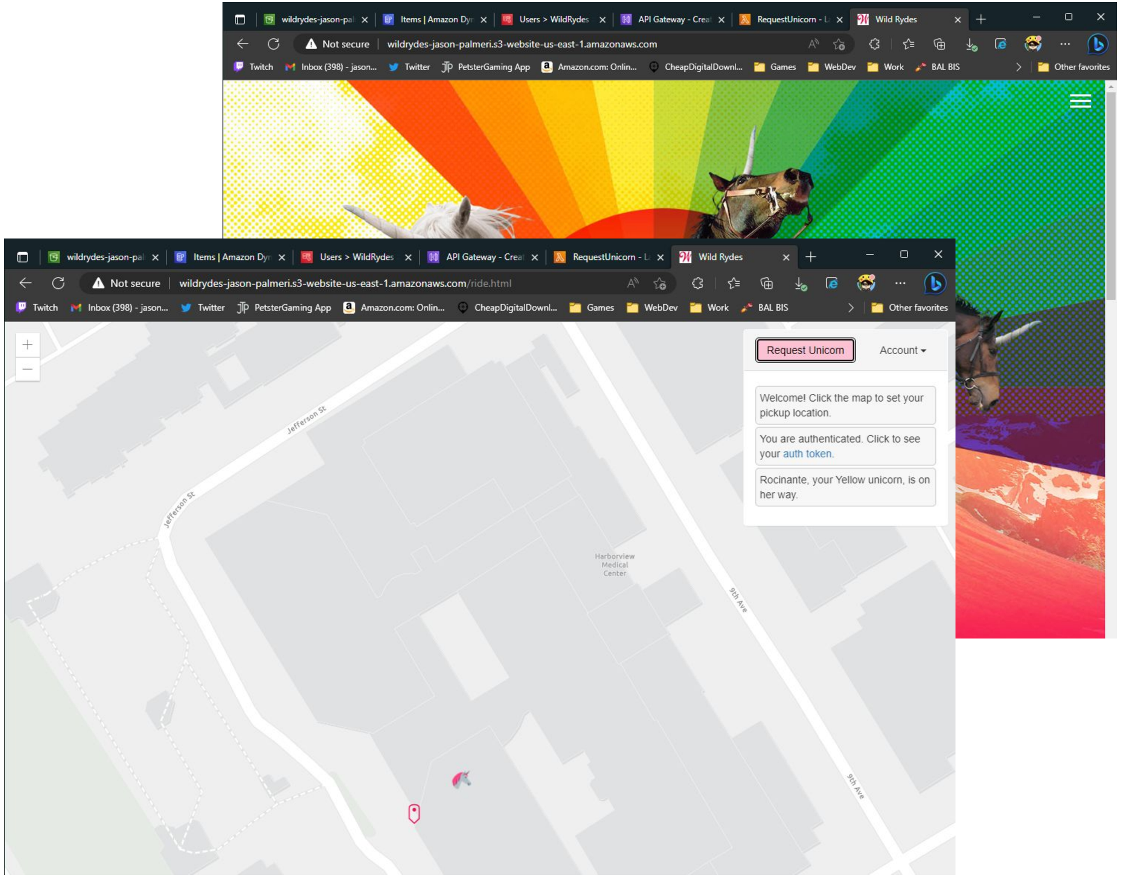
3. Finally we record the data to our DynamoDB, and send data back to the client



```
▼ {RideId: "MwASj6ohEp1sUFXjpNeWCg", Unicorn: {Name: "Bucephalus", Color: "Golden", Gender: "Male"},…}
    Eta: "30 seconds"
    RideId: "MwASj6ohEp1sUFXjpNeWCg"
    Rider: "a9c20fb6-b75c-40de-80e4-a20ccbd9efe9"
  ▶ Unicorn: {Name: "Bucephalus", Color: "Golden", Gender: "Male"}
```

The Result

With these AWS services working together we can deliver a polished web application with user authentication, API gateways and database storage, all with pay-as-you-go pricing that will allow us to expand as needed without extra hassle on the development team.



Not secure | wildrydes-jason-palmeri.s3-website-us-east-1.amazonaws.com

Not secure | wildrydes-jason-palmeri.s3-website-us-east-1.amazonaws.com/ride.html

Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Rocinante, your Yellow unicorn, is on her way.

Architecture Design Choices

Most of the choices made for WildRydes is based around the reliability and scalability of the platform. We chose AWS S3 to host our website and it's files because it is easily accessible using the cloud, and can also be scaled up or down based on the needs WildRydes has and will have.

We made the choice to use the combination of DynamoDB and Cognito because of it's easy setup, and scalability. Cognito made it simple to setup a user management system. Similarly DynamoDB allowed us to create database tables easily to view our ride information.

Finally we used Lambda and the API Gateway to create functions and routes to these functions. The API Gateway makes it easy to create, secure, and monitor API Gateways, as having these gateways unsecure could be troublesome.

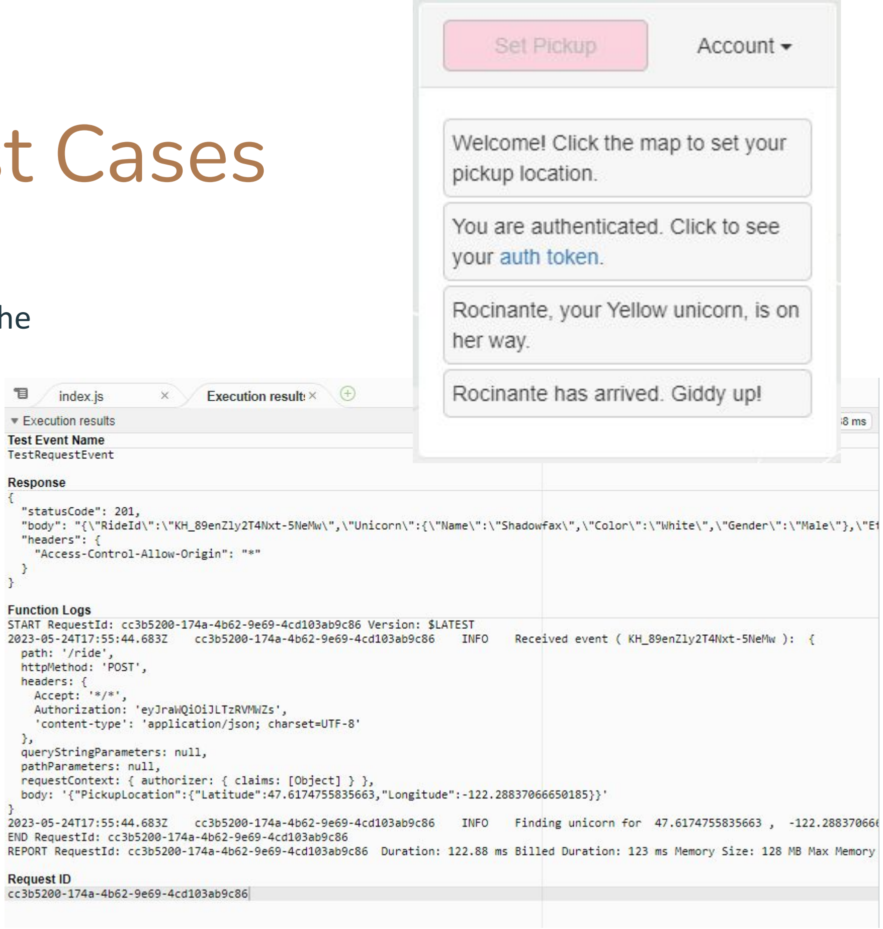
All choices were made with the scale of the current company, and the future company in mind, as well a pay-as-you-go mindset to help get the startup going.

Quality Assurance Test Cases

The team ran through several Test Cases to ensure that the product functions properly on all ends.

A few tests that were ran:

1. Validated/Unvalidated user access
2. RequestUnicorn functionality test
3. Registration, Login, Logout functionality test
4. Web app access tests, page loading tests



Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Rocinante, your Yellow unicorn, is on her way.

Rocinante has arrived. Giddy up!

▾ Execution results

Test Event Name

```
TestRequestEvent
```

Response

```
{
  "statusCode": 201,
  "body": "{\"RideId\":\"KH_89enZly2T4Nxt-5NeMw\",\"Unicorn\":{\"Name\":\"Shadowfax\",\"Color\":\"White\",\"Gender\":\"Male\"},\"Et
  "headers": {
    "Access-Control-Allow-Origin": "*"
  }
}
```

Function Logs

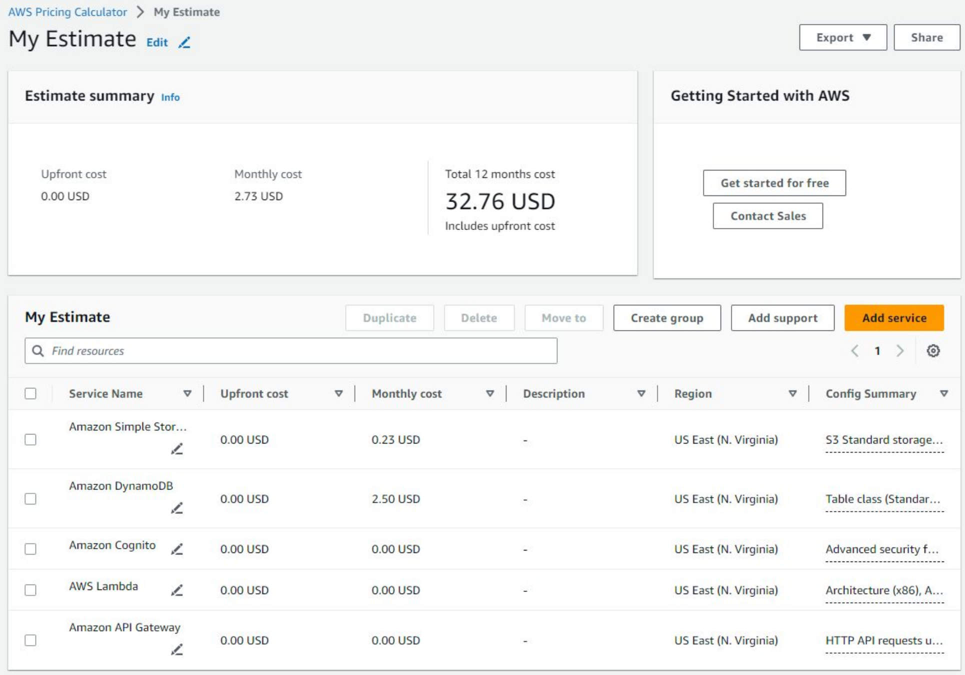
```
START RequestId: cc3b5200-174a-4b62-9e69-4cd103ab9c86 Version: $LATEST
2023-05-24T17:55:44.683Z	cc3b5200-174a-4b62-9e69-4cd103ab9c86	INFO	Received event ( KH_89enZly2T4Nxt-5NeMw ):  {
  path: '/ride',
  httpMethod: 'POST',
  headers: {
    Accept: '*/*',
    Authorization: 'eyJraWQiOiJLTzRVMWZs',
    'content-type': 'application/json; charset=UTF-8'
  },
  queryStringParameters: null,
  pathParameters: null,
  requestContext: { authorizer: { claims: [Object] } },
  body: '{"PickupLocation":{"Latitude":47.6174755835663,"Longitude":-122.28837066650185}}'
}
2023-05-24T17:55:44.683Z	cc3b5200-174a-4b62-9e69-4cd103ab9c86	INFO	Finding unicorn for  47.6174755835663 ,  -122.28837066
END RequestId: cc3b5200-174a-4b62-9e69-4cd103ab9c86
REPORT RequestId: cc3b5200-174a-4b62-9e69-4cd103ab9c86  Duration: 122.88 ms Billed Duration: 123 ms Memory Size: 128 MB Max Memory
```

Request ID

```
cc3b5200-174a-4b62-9e69-4cd103ab9c86
```

Cost Analysis and Breakdown - Startup

For WildRydes startup costs we are looking at around $2.73 per month, with 100 monthly users requesting 10 rides each. As you can see the cost is minimal for the year at $32.76. The biggest cost is in the DynamoDB where we are storing our customer transaction data. The next biggest cost is the S3 service, hosting our web application and it's files.



My Estimate

Estimate summary

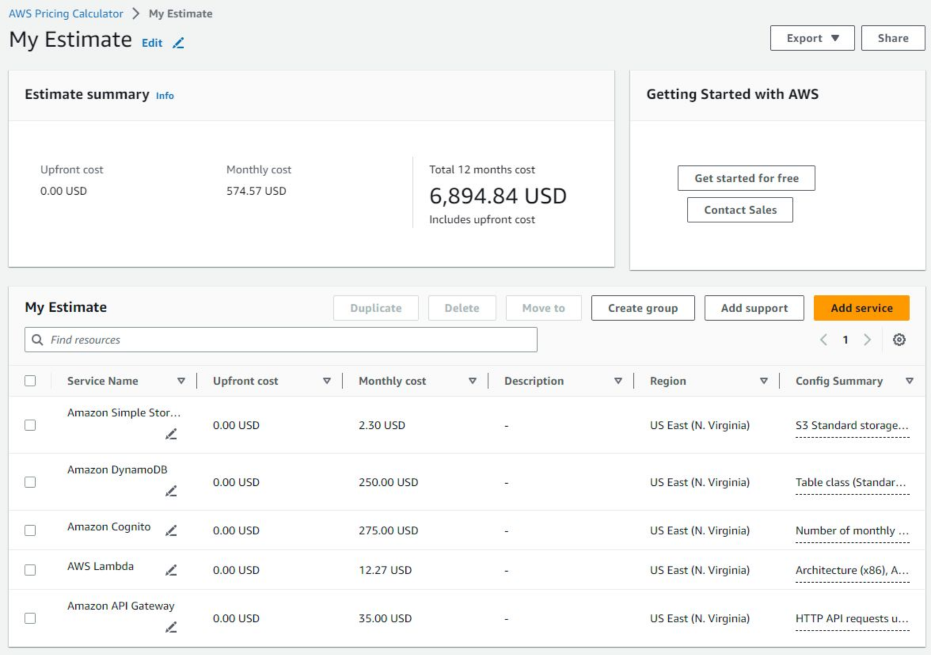
Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	2.73 USD	32.76 USD Includes upfront cost

Getting Started with AWS

My Estimate

□	Service Name	Upfront cost	Monthly cost	Description	Region	Config Summary
□	Amazon Simple Stor...	0.00 USD	0.23 USD	-	US East (N. Virginia)	S3 Standard storage...
□	Amazon DynamoDB	0.00 USD	2.50 USD	-	US East (N. Virginia)	Table class (Standar...
□	Amazon Cognito	0.00 USD	0.00 USD	-	US East (N. Virginia)	Advanced security f...
□	AWS Lambda	0.00 USD	0.00 USD	-	US East (N. Virginia)	Architecture (x86), A...
□	Amazon API Gateway	0.00 USD	0.00 USD	-	US East (N. Virginia)	HTTP API requests u...

Cost Analysis and Breakdown - Success



My Estimate

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	574.57 USD	6,894.84 USD Includes upfront cost

Getting Started with AWS

My Estimate

☐	Service Name	Upfront cost	Monthly cost	Description	Region	Config Summary
☐	Amazon Simple Stor...	0.00 USD	2.30 USD	-	US East (N. Virginia)	S3 Standard storage...
☐	Amazon DynamoDB	0.00 USD	250.00 USD	-	US East (N. Virginia)	Table class (Standar...
☐	Amazon Cognito	0.00 USD	275.00 USD	-	US East (N. Virginia)	Number of monthly ...
☐	AWS Lambda	0.00 USD	12.27 USD	-	US East (N. Virginia)	Architecture (x86), A...
☐	Amazon API Gateway	0.00 USD	35.00 USD	-	US East (N. Virginia)	HTTP API requests u...

As WildRydes expands and sees over 1 million monthly active users, requesting at least 10 rides per month we can see over a 20000% increase in yearly costs, at $6894.84. With a larger user set and more rides being requested we have to increase the size of both our DynamoDB and Cognito systems, which both consume around $250 per month. Some services stay at a lower price such as the S3, which doesn't need to be expanded as much as the other services, and the Lambda service, which is only handling a small function.