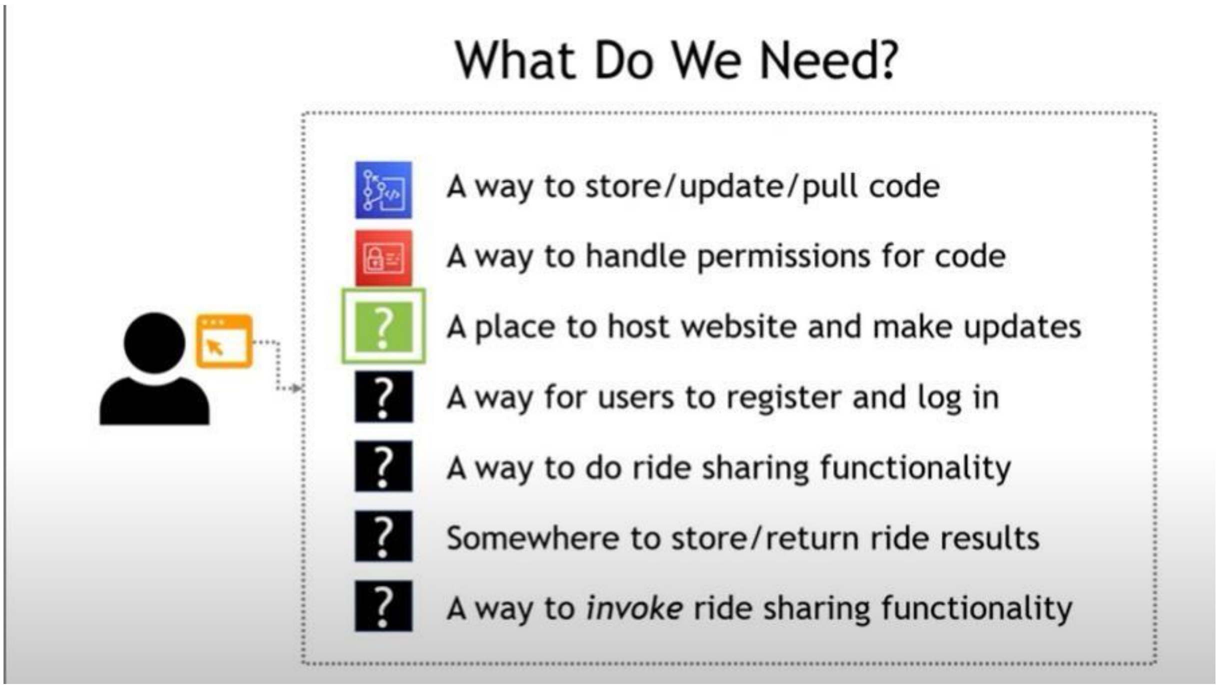
Building a Serverless Web Application by Sheelampally Sai Shiva

with AWS Lambda, Amazon API Gateway, AWS Amplify, Amazon DynamoDB, and Amazon Cognito



What Do We Need?

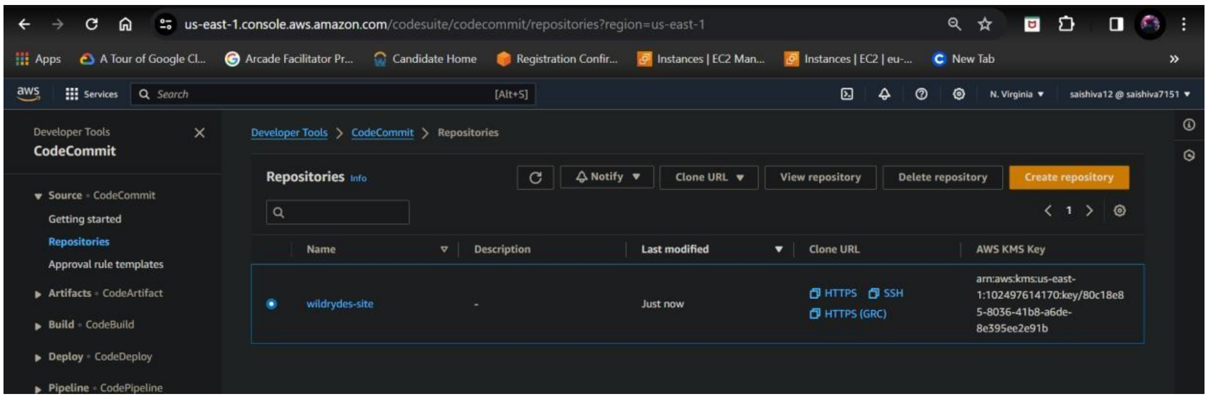
- A way to store/update/pull code
- A way to handle permissions for code
- A place to host website and make updates
- A way for users to register and log in
- A way to do ride sharing functionality
- Somewhere to store/return ride results
- A way to *invoke* ride sharing functionality

- AWS Amplify ::Used to host the static resources.Continous deployment is supported

>>First we create a repository in CodeCommit Repo

We take the code provided in the public S3 bucket and copy it to the code repository in the created repository in the CodeCommit.

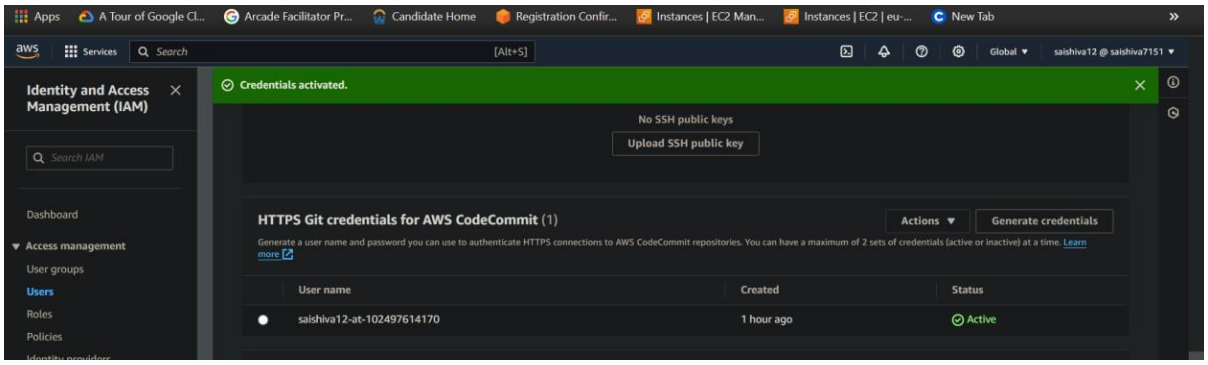
- First we create a repository in CodeCommit Repo



Name	Description	Last modified	Clone URL	AWS KMS Key
wildrydes-site	-	Just now	HTTPS SSH HTTPS (GRC)	arn:aws:kms:us-east-1:102497614170:key/80c18e85-8036-41b8-a6de-8e395ee2e91b

- Then we add the policy AWSCodeCommitPowerUser to allow access for making changes to the code commit.
- To allow https connections to CodeCommit we need to create the Git credentials for IAM user
  >making of credentials; HTTPS Git credentials for AWS CodeCommit

saishiva12-at-10********0 PD :: DMLbDU+aKlFkML1Uwjn************es8=


Credentials activated.

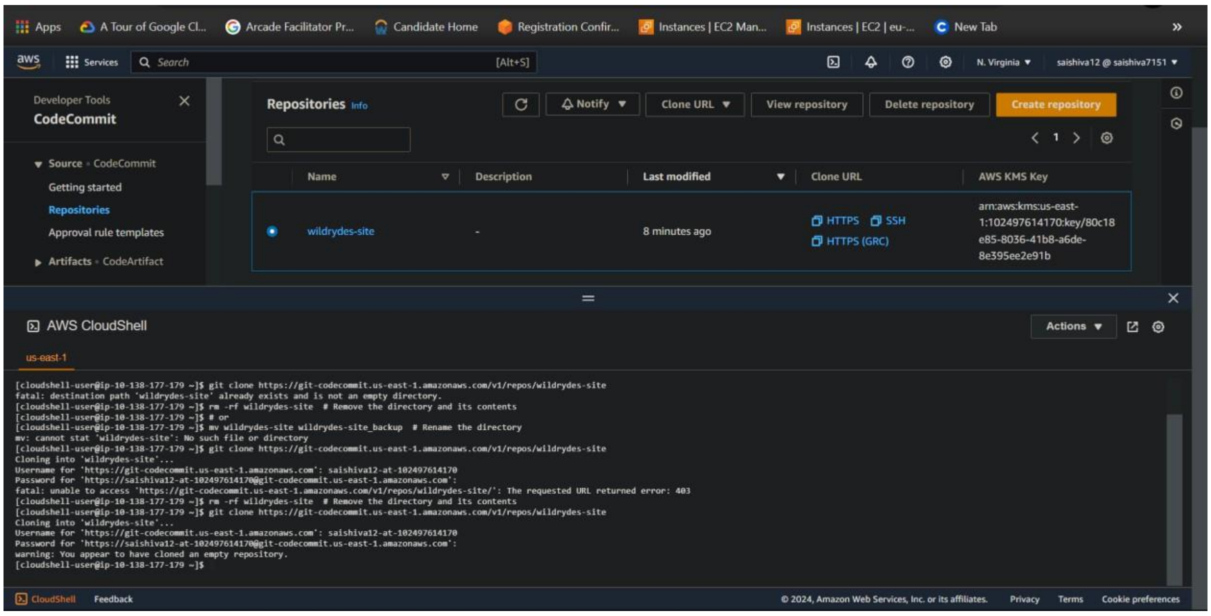
No SSH public keys

HTTPS Git credentials for AWS CodeCommit (1)

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. Learn more

User name	Created	Status
saishiva12-at-102497614170	1 hour ago	Active

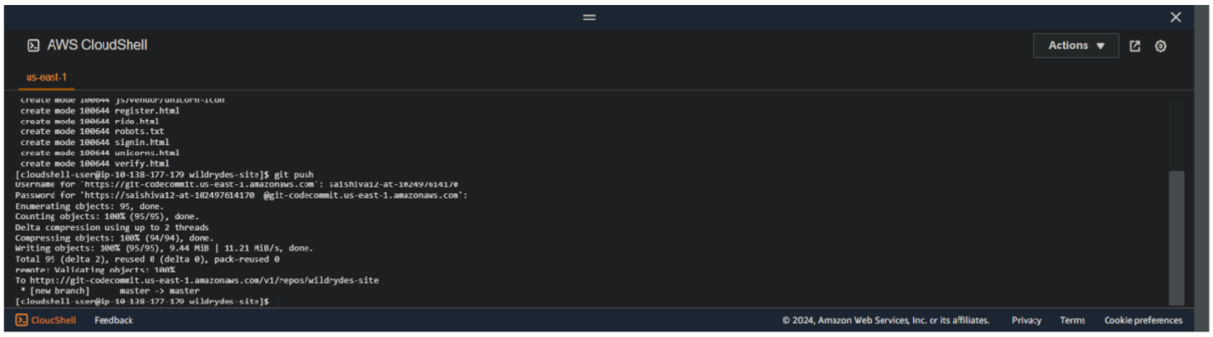
>now clone for the repository(create an empty folder for future code)


Repositories

Name	Description	Last modified	Clone URL	AWS KMS Key
wildrydes-site	-	8 minutes ago	HTTPS SSH HTTPS (GRC)	arn:aws:kms:us-east-1:102497614170:key/80c18e85-8036-41b8-a6de-8e395ee2e91b

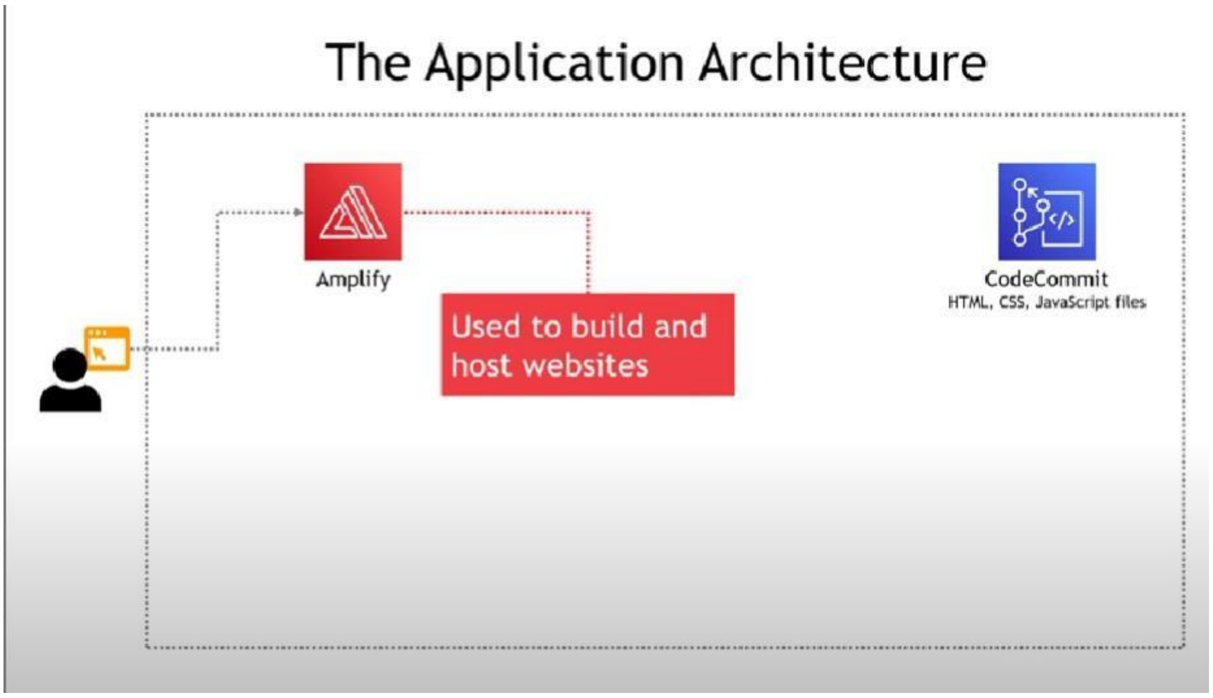
```
[cloudshell-user@ip-10-138-177-179 ~]$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
fatal: destination path 'wildrydes-site' already exists and is not an empty directory.
[cloudshell-user@ip-10-138-177-179 ~]$ rm -rf wildrydes-site  # Remove the directory and its contents
[cloudshell-user@ip-10-138-177-179 ~]$ # or
[cloudshell-user@ip-10-138-177-179 ~]$ mv wildrydes-site wildrydes-site_backup  # Rename the directory
mv: cannot stat 'wildrydes-site': No such file or directory
[cloudshell-user@ip-10-138-177-179 ~]$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
Cloning into 'wildrydes-site'...
Username for 'https://git-codecommit.us-east-1.amazonaws.com': saishiva12-at-102497614170
Password for 'https://saishiva12-at-102497614170@git-codecommit.us-east-1.amazonaws.com':
fatal: unable to access 'https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site/': The requested URL returned error: 403
[cloudshell-user@ip-10-138-177-179 ~]$ rm -rf wildrydes-site  # Remove the directory and its contents
[cloudshell-user@ip-10-138-177-179 ~]$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
Cloning into 'wildrydes-site'...
Username for 'https://git-codecommit.us-east-1.amazonaws.com': saishiva12-at-102497614170
Password for 'https://saishiva12-at-102497614170@git-codecommit.us-east-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
[cloudshell-user@ip-10-138-177-179 ~]$
```

>>Successfully pushed the code into git repository

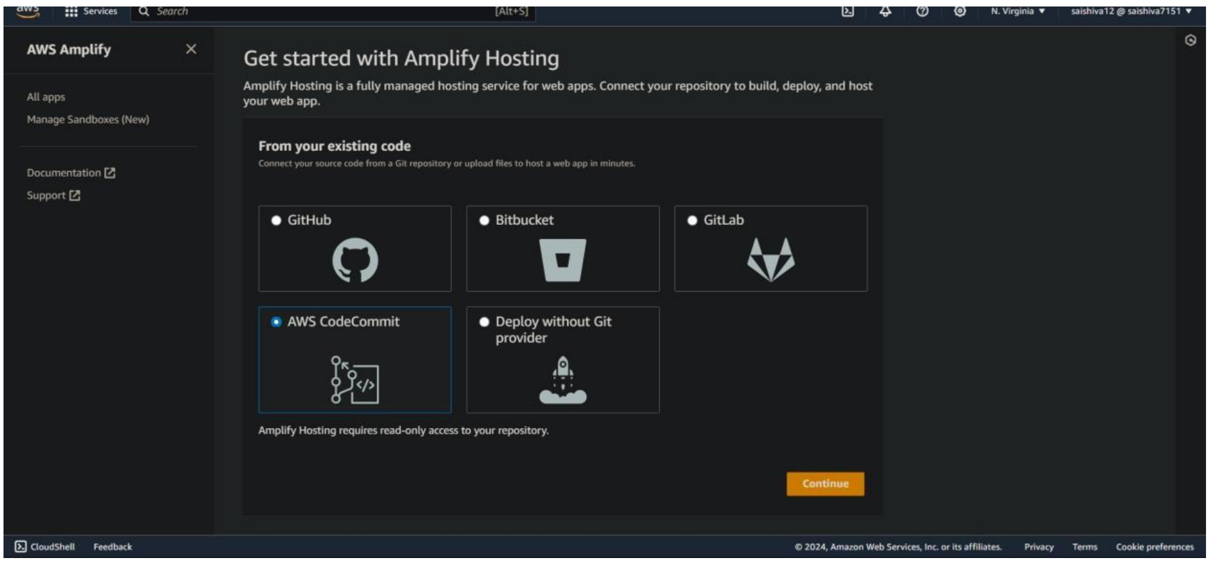

```
 create mode 100644 js/vendor/unicorn-icon
 create mode 100644 register.html
 create mode 100644 ride.html
 create mode 100644 robots.txt
 create mode 100644 signin.html
 create mode 100644 unicorns.html
 create mode 100644 verify.html
[cloudshell-user@ip-10-138-177-179 wildrydes-site]$ git push
Username for 'https://git-codecommit.us-east-1.amazonaws.com': saishiva12-at-102497614170
Password for 'https://saishiva12-at-102497614170  @git-codecommit.us-east-1.amazonaws.com':
Enumerating objects: 95, done.
Counting objects: 100% (95/95), done.
Delta compression using up to 2 threads
Compressing objects: 100% (94/94), done.
Writing objects: 100% (95/95), 9.44 MiB | 11.21 MiB/s, done.
Total 95 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wildrydes-site
 * [new branch]      master -> master
[cloudshell-user@ip-10-138-177-179 wildrydes-site]$
```

- Allow a place to host website amd make updates

Amplify is the service where we build and host the websites


The Application Architecture
Amplify
Used to build and host websites
CodeCommit
HTML, CSS, JavaScript files

For hosting the website we are choosing the existing source code from code commit



Get started with Amplify Hosting

Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

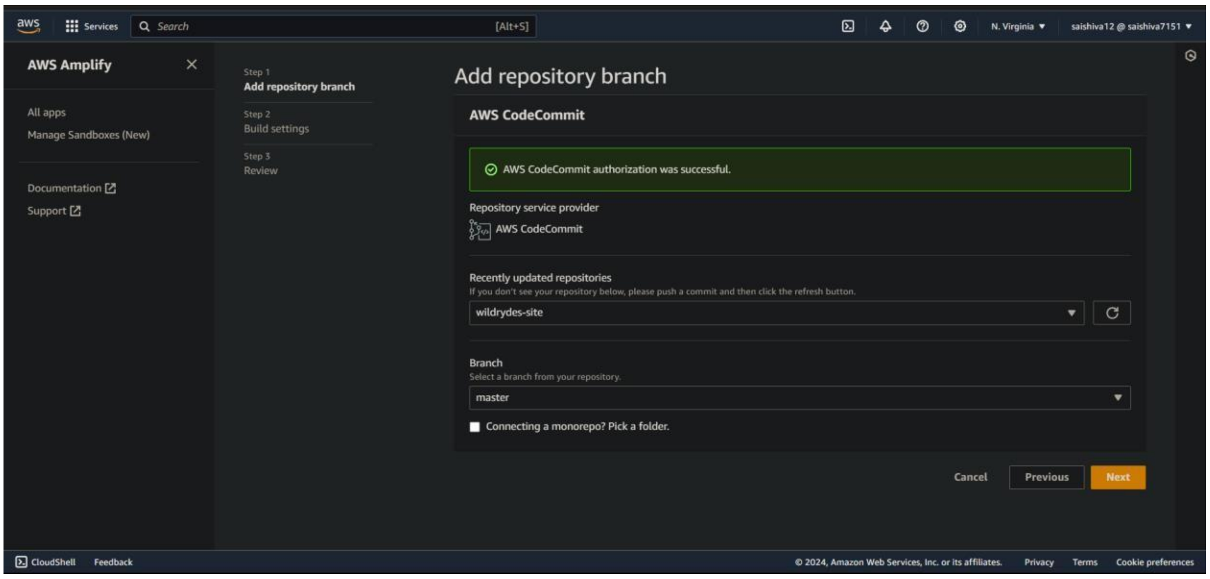
From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.

- GitHub
- Bitbucket
- GitLab
- [✓ AWS CodeCommit]
- Deploy without Git provider

Amplify Hosting requires read-only access to your repository.

Continue



Add repository branch

AWS CodeCommit

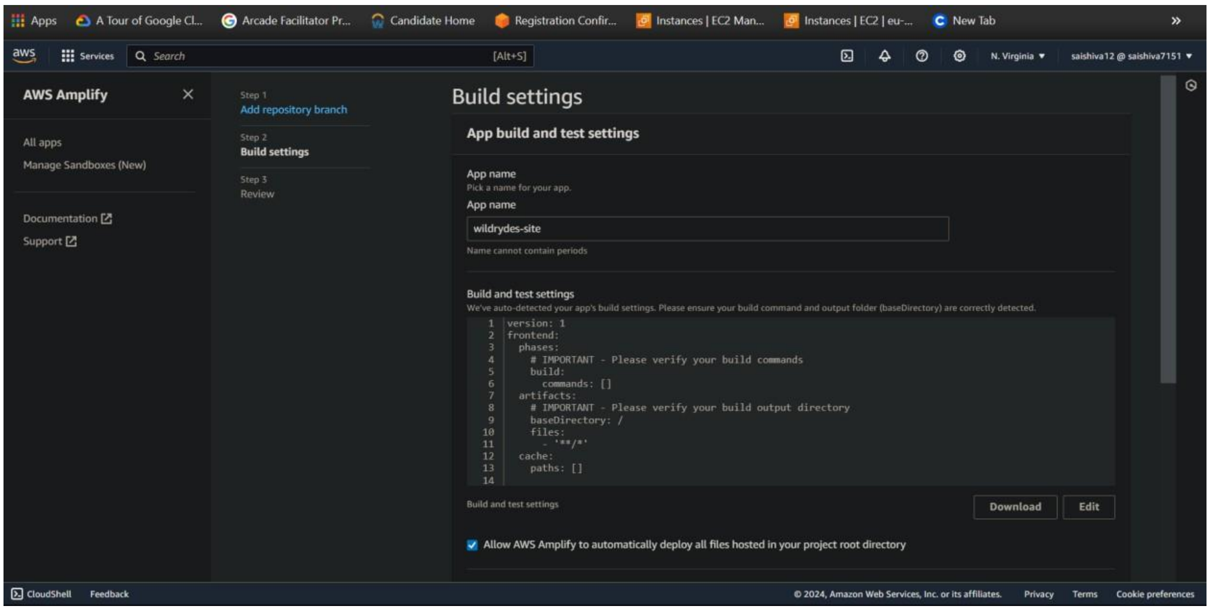
AWS CodeCommit authorization was successful.

Repository service provider
AWS CodeCommit

Recently updated repositories
If you don't see your repository below, please push a commit and then click the refresh button.
wildrydes-site

Branch
Select a branch from your repository.
master

☐ Connecting a monorepo? Pick a folder.



Build settings

App build and test settings

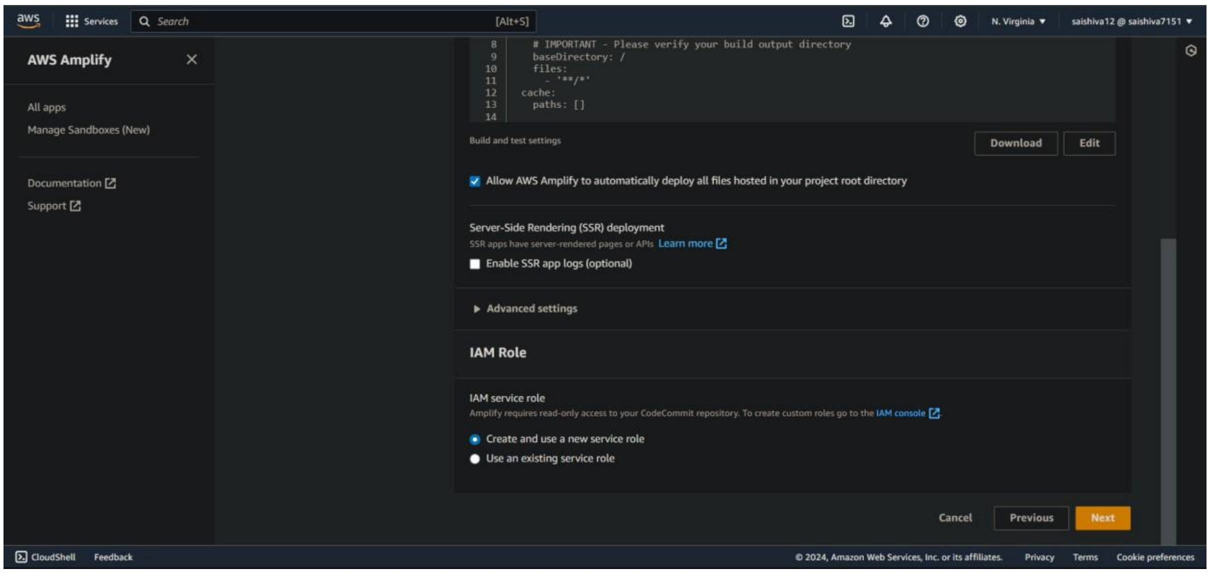
App name
Pick a name for your app.
App name
wildrydes-site
Name cannot contain periods

Build and test settings
We've auto-detected your app's build settings. Please ensure your build command and output folder (baseDirectory) are correctly detected.

```
version: 1
frontend:
  phases:
    # IMPORTANT - Please verify your build commands
    build:
      commands: []
  artifacts:
    # IMPORTANT - Please verify your build output directory
    baseDirectory: /
    files:
      - '**/*'
  cache:
    paths: []
```

Build and test settings

☑ Allow AWS Amplify to automatically deploy all files hosted in your project root directory



```
    # IMPORTANT - Please verify your build output directory
    baseDirectory: /
    files:
      - '**/*'
  cache:
    paths: []
```

Build and test settings

☑ Allow AWS Amplify to automatically deploy all files hosted in your project root directory

Server-Side Rendering (SSR) deployment
SSR apps have server-rendered pages or APIs Learn more

☐ Enable SSR app logs (optional)

▸ Advanced settings

IAM Role

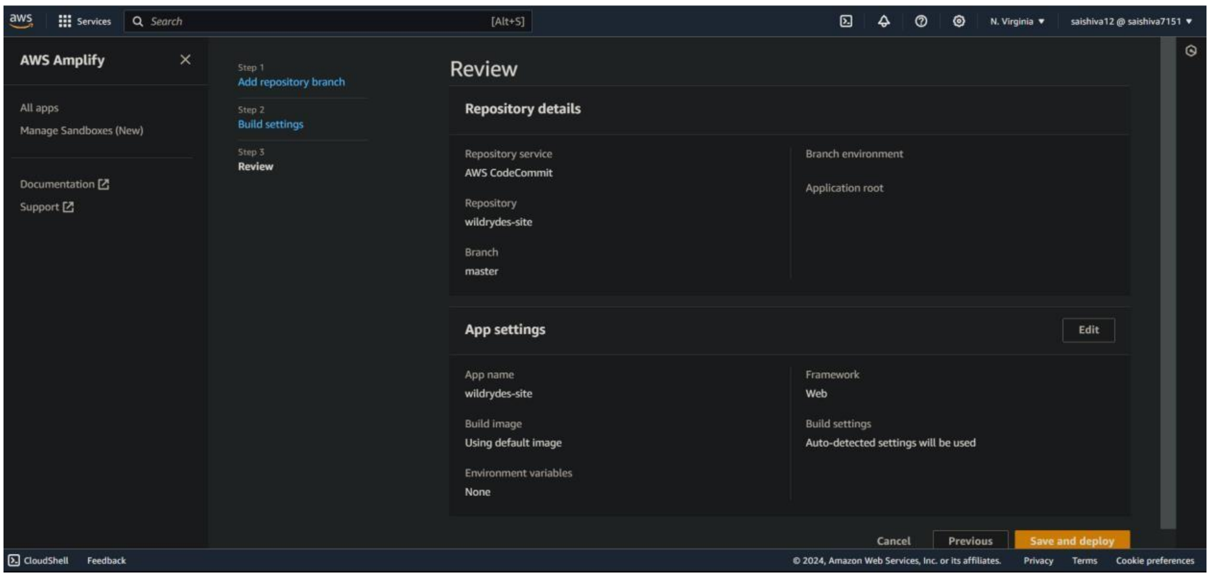
IAM service role
Amplify requires read-only access to your CodeCommit repository. To create custom roles go to the IAM console

● Create and use a new service role
○ Use an existing service role

>choose allow automatic deploy of all files in Amplify

>and choos e Create and use a new service role in IAM role

Review ::



AWS Amplify

All apps
Manage Sandboxes (New)
Documentation
Support

Step 1
Add repository branch
Step 2
Build settings
Step 3
Review

Review

Repository details

Repository service
AWS CodeCommit
Repository
wildrydes-site
Branch
master
Branch environment
Application root

App settings

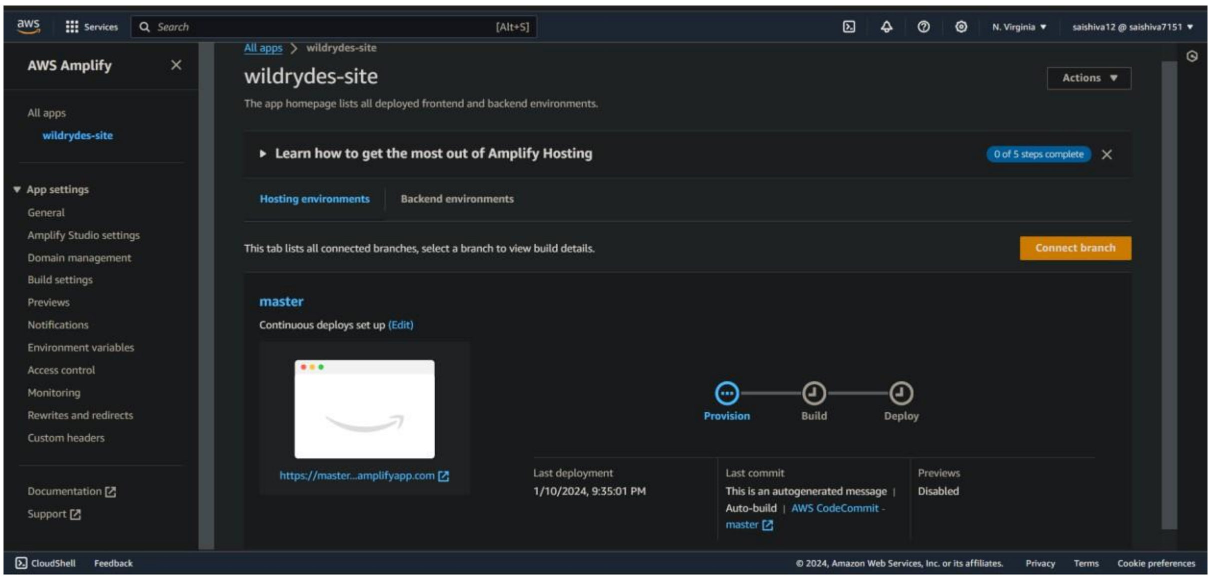
Edit

App name
wildrydes-site
Build image
Using default image
Environment variables
None
Framework
Web
Build settings
Auto-detected settings will be used

Cancel Previous Save and deploy

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now we can watch the progress,it is a serverless as it doesn't need to set up EC2 instance for hosting the website



AWS Amplify

All apps
wildrydes-site
App settings
General
Amplify Studio settings
Domain management
Build settings
Previews
Notifications
Environment variables
Access control
Monitoring
Rewrites and redirects
Custom headers
Documentation
Support

All apps > wildrydes-site

wildrydes-site

The app homepage lists all deployed frontend and backend environments.

Actions

Learn how to get the most out of Amplify Hosting 0 of 5 steps complete

Hosting environments Backend environments

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

Connect branch

master
Continuous deploys set up (Edit)

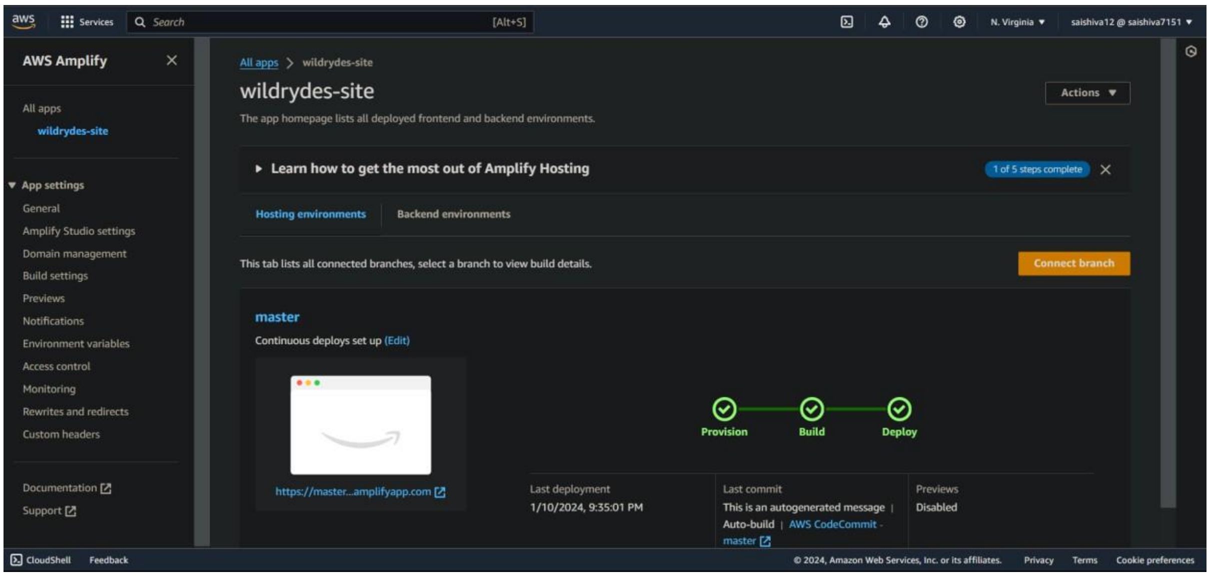
https://master...amplifyapp.com

Provision Build Deploy

Last deployment
1/10/2024, 9:35:01 PM
Last commit
This is an autogenerated message | Auto-build | AWS CodeCommit - master
Previews
Disabled

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The app is success fully deployed



wildrydes-site

The app homepage lists all deployed frontend and backend environments.

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

master

Continuous deploys set up (Edit)

https://master...amplifyapp.com

Last deployment: 1/10/2024, 9:35:01 PM

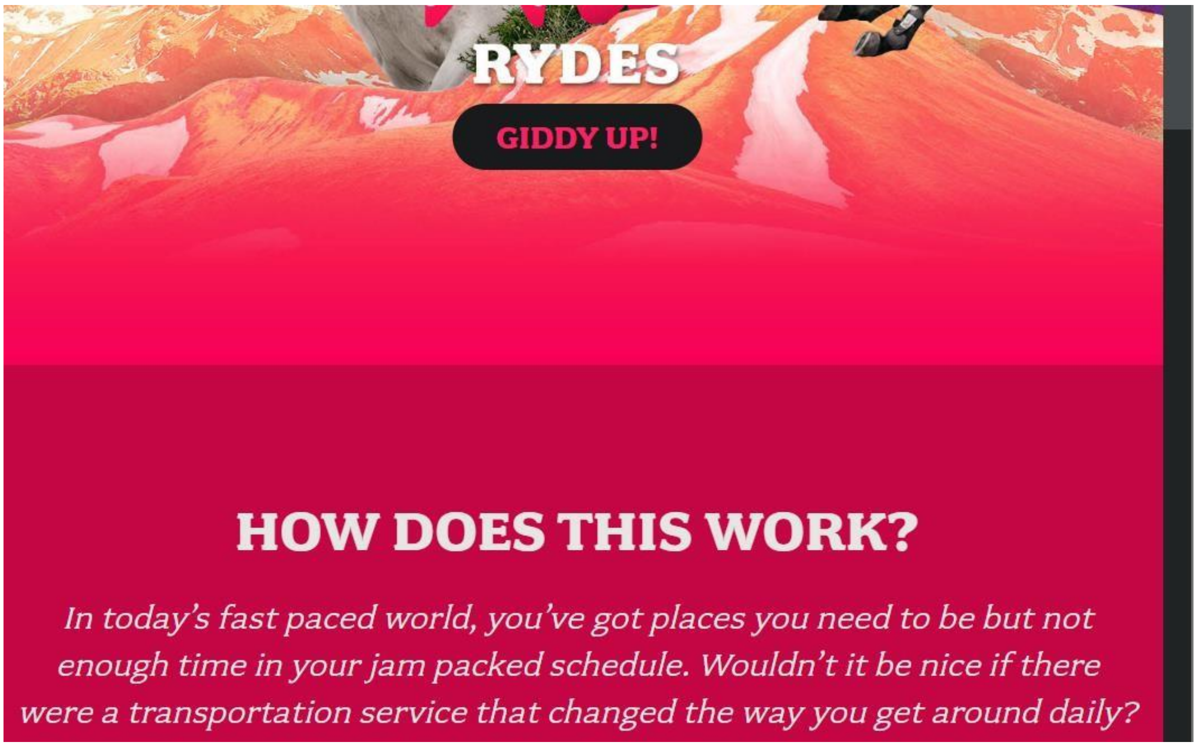
Last commit: This is an autogenerated message | Auto-build | AWS CodeCommit - master

Previews: Disabled

Now let us check if the contnious deployment is working or not .To check we have to make changes to the HTML code and check whether is it updating or not.

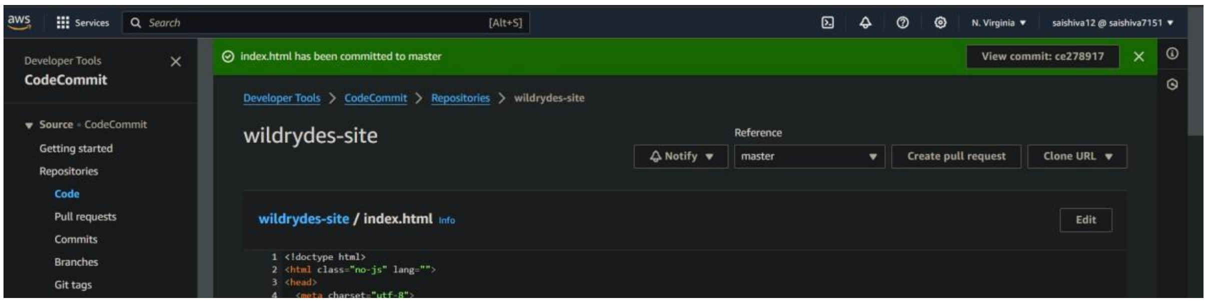
Before changing :

Notice the heading “HOW DOES THIS WORK?” I am going to change It to “HOW DOES THIS **THING** WORK?”



HOW DOES THIS WORK?

In today’s fast paced world, you’ve got places you need to be but not enough time in your jam packed schedule. Wouldn’t it be nice if there were a transportation service that changed the way you get around daily?



index.html has been committed to master

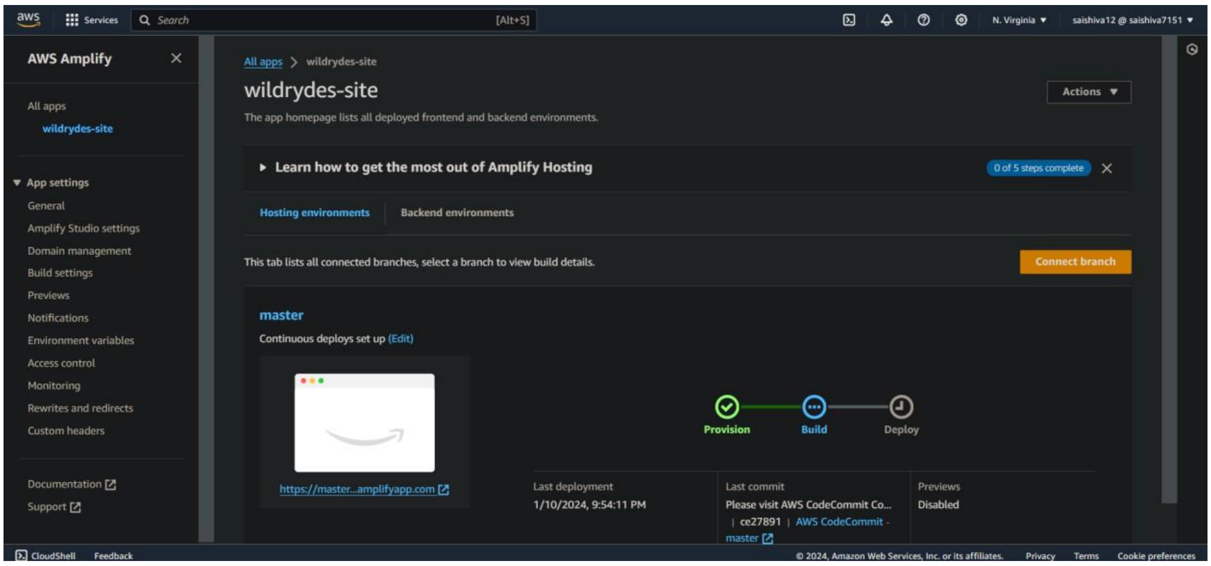
Developer Tools > CodeCommit > Repositories > wildrydes-site

wildrydes-site

wildrydes-site / index.html

```
<!doctype html>
<html class="no-js" lang="">
<head>
  <meta charset="utf-8">
```

>The changes are being commited



All apps > wildrydes-site

wildrydes-site

The app homepage lists all deployed frontend and backend environments.

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

master

Continuous deploys set up (Edit)

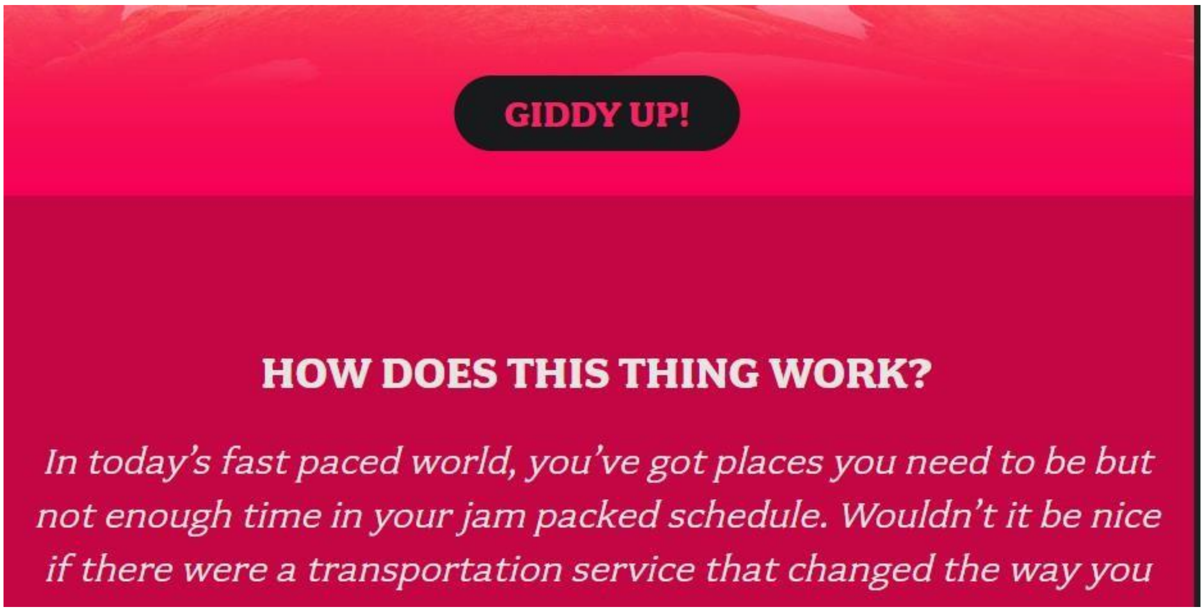
https://master...amplifyapp.com

Last deployment
1/10/2024, 9:54:11 PM

Last commit
Please visit AWS CodeCommit Co...
| ce27891 | AWS CodeCommit - master

Previews
Disabled

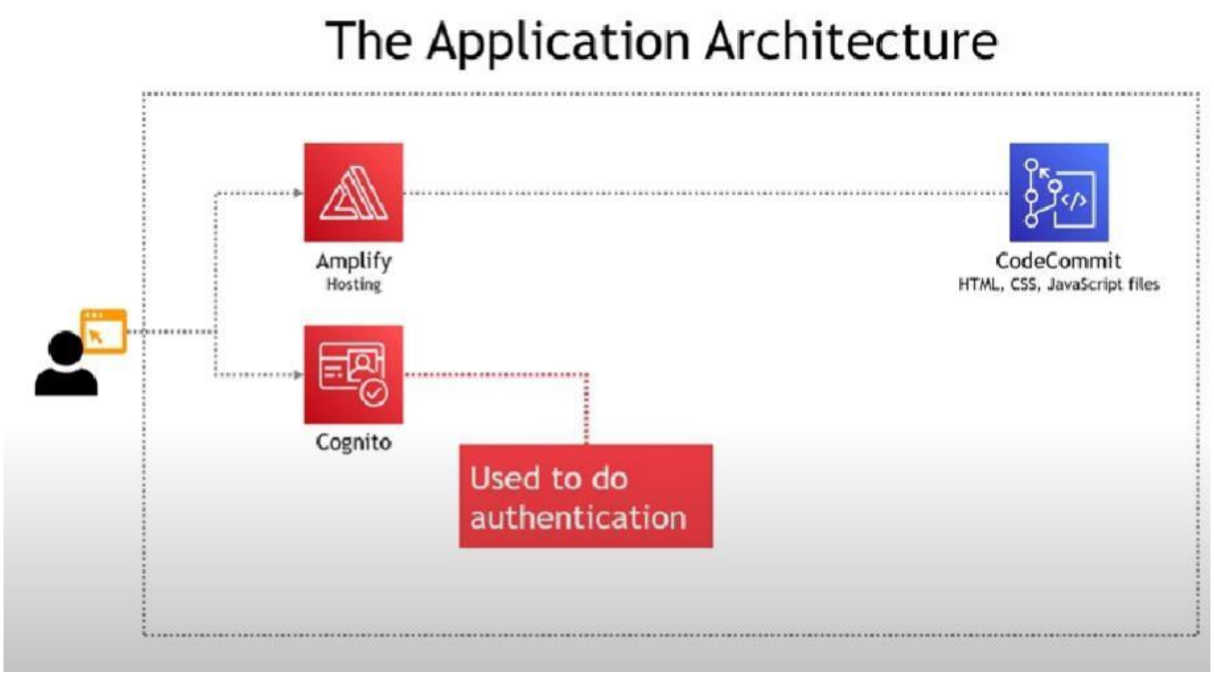
>> The change Is success fully shown



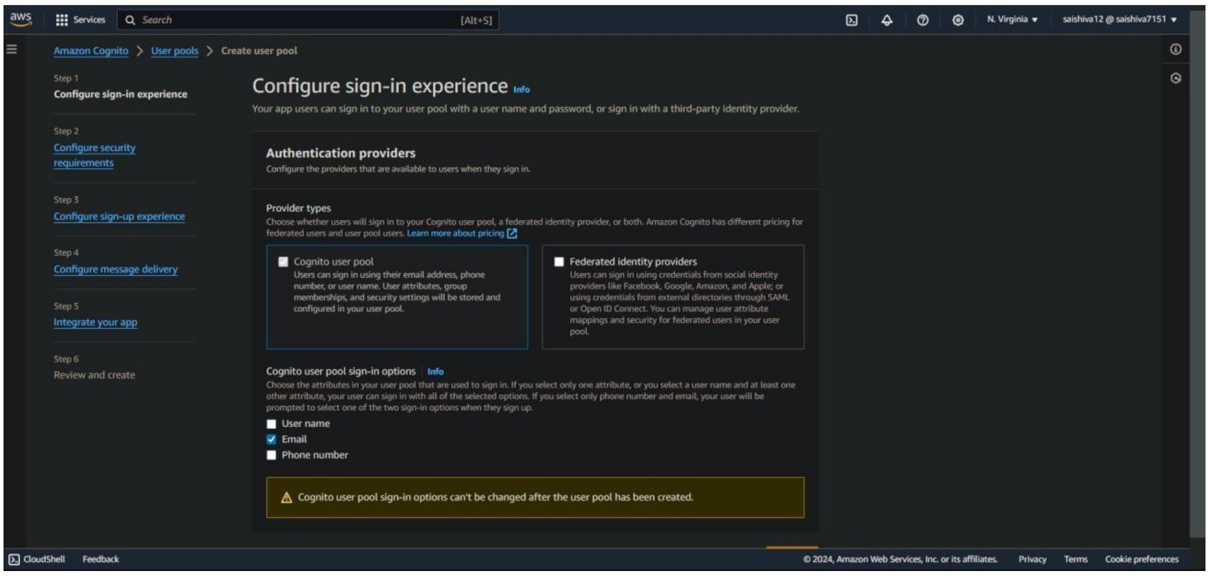
HOW DOES THIS THING WORK?

In today's fast paced world, you've got places you need to be but not enough time in your jam packed schedule. Wouldn't it be nice if there were a transportation service that changed the way you

- 4
  Now we need a way to users to register and log in
  For this we use Cognito : Which is used for authentication


The Application Architecture
Amplify
Hosting
CodeCommit
HTML, CSS, JavaScript files
Cognito
Used to do authentication

Creating a user pool in cognito


aws
Services
Search
[Alt+S]
N. Virginia
saishiva12 @ saishiva7151
Amazon Cognito > User pools > Create user pool
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure sign-in experience Info

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

Authentication providers

Configure the providers that are available to users when they sign in.

Provider types

Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. Learn more about pricing

- [✓] Cognito user pool
  Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.
- [ ] Federated identity providers
  Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

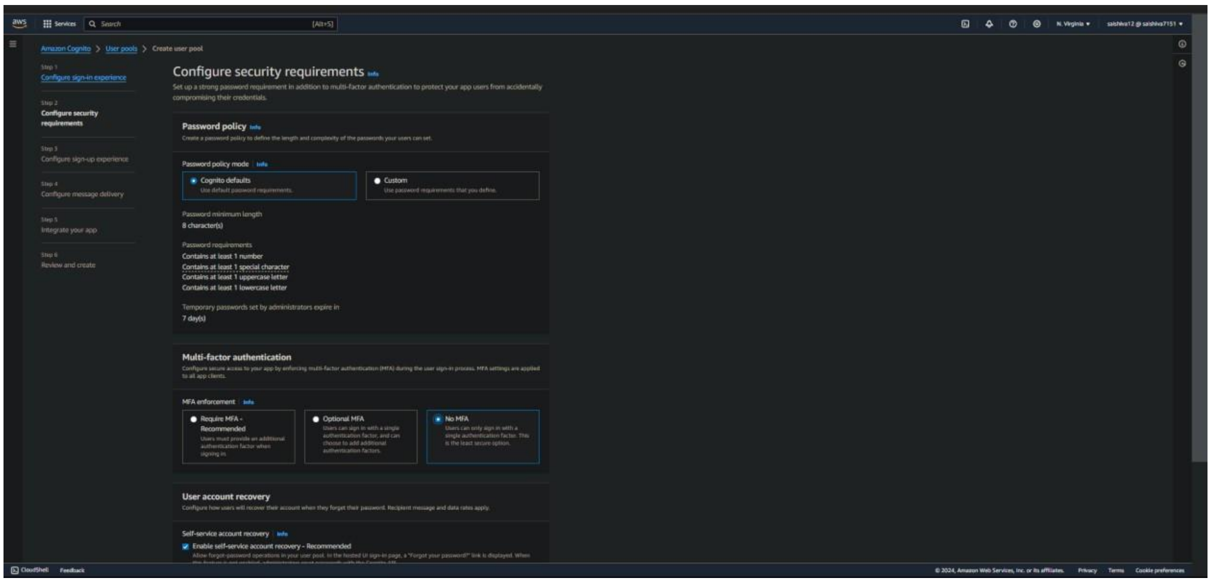
Cognito user pool sign-in options Info

Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

- [ ] User name
- [✓] Email
- [ ] Phone number

⚠ Cognito user pool sign-in options can't be changed after the user pool has been created.

Choose No MFA


Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure security requirements Info

Set up a strong password requirement in addition to multi-factor authentication to protect your app users from accidentally compromising their credentials.

Password policy Info
Create a password policy to define the length and complexity of the passwords your users can set.

Password policy mode | Info
[✓ Cognito defaults] Use default password requirements.
Custom — Use password requirements that you define.

Password minimum length
8 character(s)

Password requirements
Contains at least 1 number
Contains at least 1 special character
Contains at least 1 uppercase letter
Contains at least 1 lowercase letter

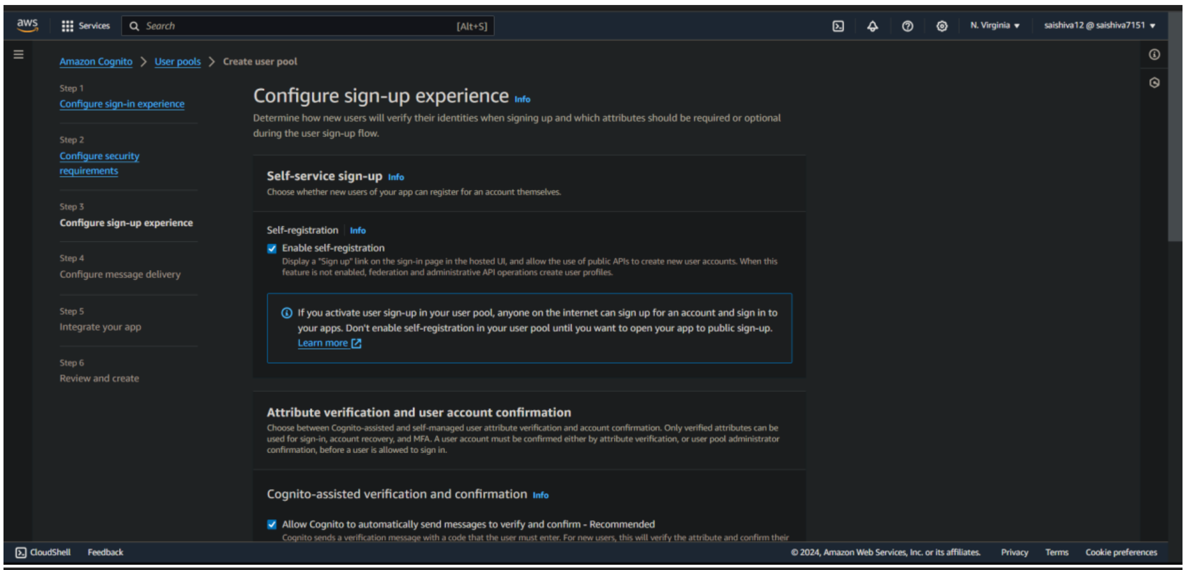
Temporary passwords set by administrators expire in
7 day(s)

Multi-factor authentication
Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

MFA enforcement | Info
Require MFA - Recommended — Users must provide an additional authentication factor when signing in.
Optional MFA — Users can sign in with a single authentication factor, and can choose to add additional authentication factors.
[✓ No MFA] Users can only sign in with a single authentication factor. This is the least secure option.

User account recovery
Configure how users will recover their account when they forget their password. Recipient message and data rates apply.

Self-service account recovery | Info
[✓] Enable self-service account recovery - Recommended
Allow forgot-password operations in your user pool. In the hosted UI sign-in page, a "Forgot your password?" link is displayed. When


Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure sign-up experience Info

Determine how new users will verify their identities when signing up and which attributes should be required or optional during the user sign-up flow.

Self-service sign-up Info
Choose whether new users of your app can register for an account themselves.

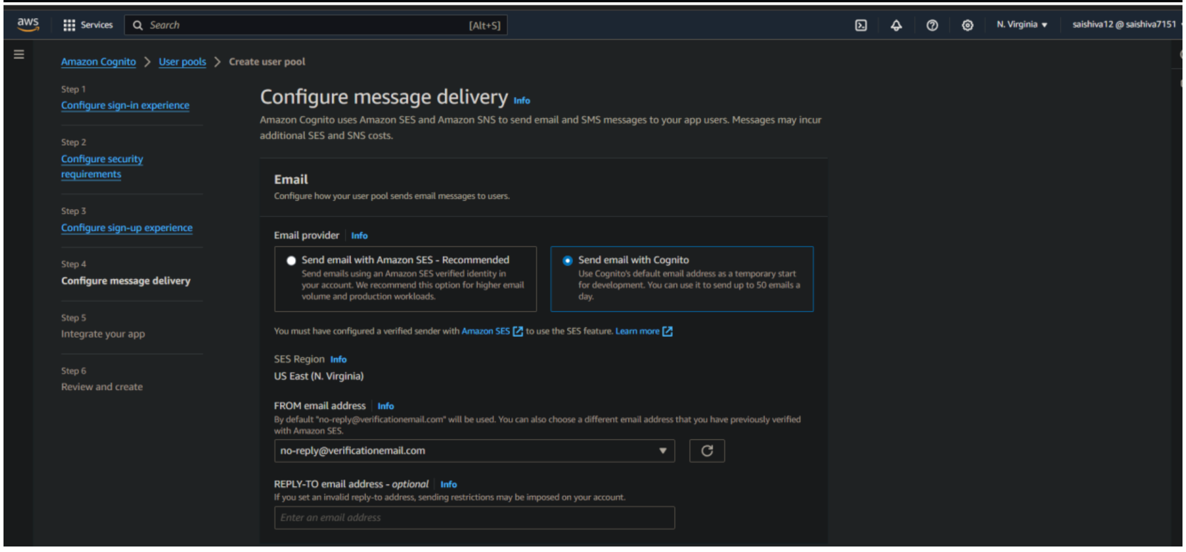
Self-registration | Info
[✓] Enable self-registration
Display a "Sign up" link on the sign-in page in the hosted UI, and allow the use of public APIs to create new user accounts. When this feature is not enabled, federation and administrative API operations create user profiles.

If you activate user sign-up in your user pool, anyone on the internet can sign up for an account and sign in to your apps. Don't enable self-registration in your user pool until you want to open your app to public sign-up.
Learn more

Attribute verification and user account confirmation
Choose between Cognito-assisted and self-managed user attribute verification and account confirmation. Only verified attributes can be used for sign-in, account recovery, and MFA. A user account must be confirmed either by attribute verification, or user pool administrator confirmation, before a user is allowed to sign in.

Cognito-assisted verification and confirmation Info

[✓] Allow Cognito to automatically send messages to verify and confirm - Recommended
Cognito sends a verification message with a code that the user must enter. For new users, this will verify the attribute and confirm their


Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure message delivery Info

Amazon Cognito uses Amazon SES and Amazon SNS to send email and SMS messages to your app users. Messages may incur additional SES and SNS costs.

Email
Configure how your user pool sends email messages to users.

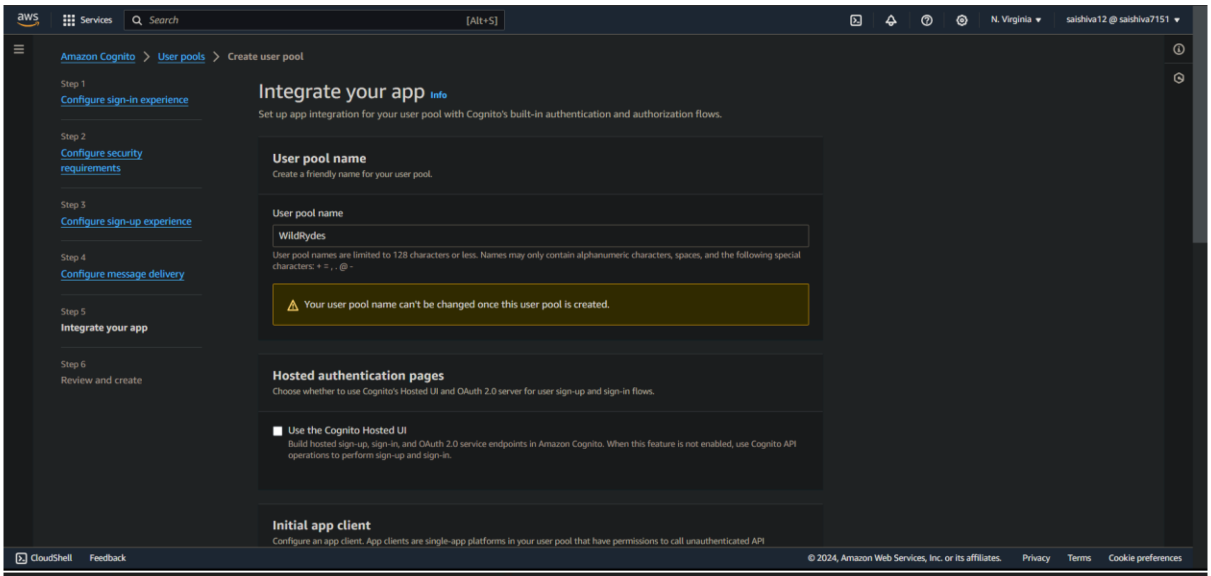
Email provider | Info
Send email with Amazon SES - Recommended — Send emails using an Amazon SES verified identity in your account. We recommend this option for higher email volume and production workloads.
[✓ Send email with Cognito] Use Cognito's default email address as a temporary start for development. You can use it to send up to 50 emails a day.

You must have configured a verified sender with Amazon SES to use the SES feature. Learn more

SES Region Info
US East (N. Virginia)

FROM email address | Info
By default "no-reply@verificationemail.com" will be used. You can also choose a different email address that you have previously verified with Amazon SES.
no-reply@verificationemail.com

REPLY-TO email address - *optional* Info
If you set an invalid reply-to address, sending restrictions may be imposed on your account.
Enter an email address


Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Integrate your app

Set up app integration for your user pool with Cognito's built-in authentication and authorization flows.

User pool name
Create a friendly name for your user pool.

User pool name: WildRydes

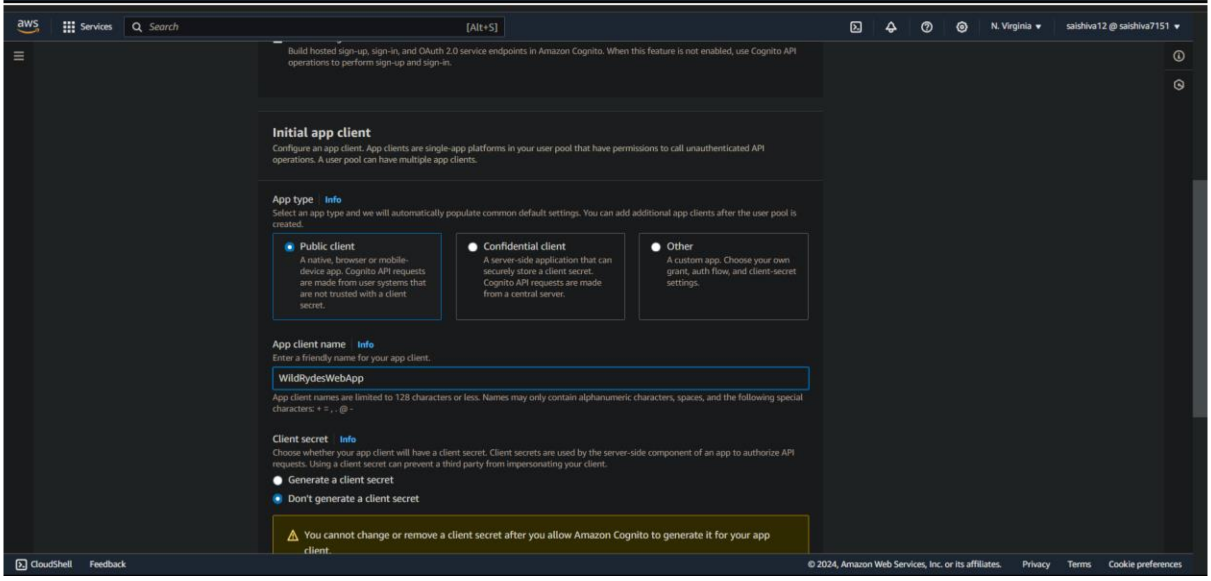
User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Your user pool name can't be changed once this user pool is created.

Hosted authentication pages
Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.

[ ] Use the Cognito Hosted UI
Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

Initial app client
Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API


Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

Initial app client
Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

App type
Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

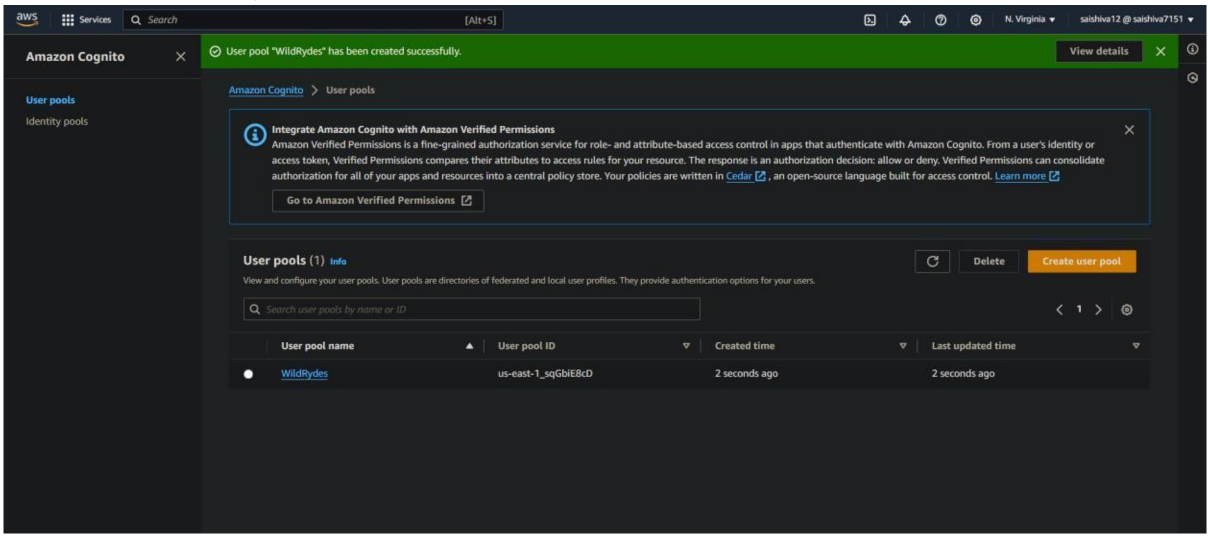
[✓ Public client] A native, browser or mobile-device app. Cognito API requests are made from user systems that are not trusted with a client secret.
Confidential client: A server-side application that can securely store a client secret. Cognito API requests are made from a central server.
Other: A custom app. Choose your own grant, auth flow, and client-secret settings.

App client name
Enter a friendly name for your app client.
WildRydesWebApp
App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Client secret
Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API requests. Using a client secret can prevent a third party from impersonating your client.
Generate a client secret
[✓ Don't generate a client secret]

You cannot change or remove a client secret after you allow Amazon Cognito to generate it for your app client.

User Pool successfully created


User pool "WildRydes" has been created successfully.

Amazon Cognito > User pools

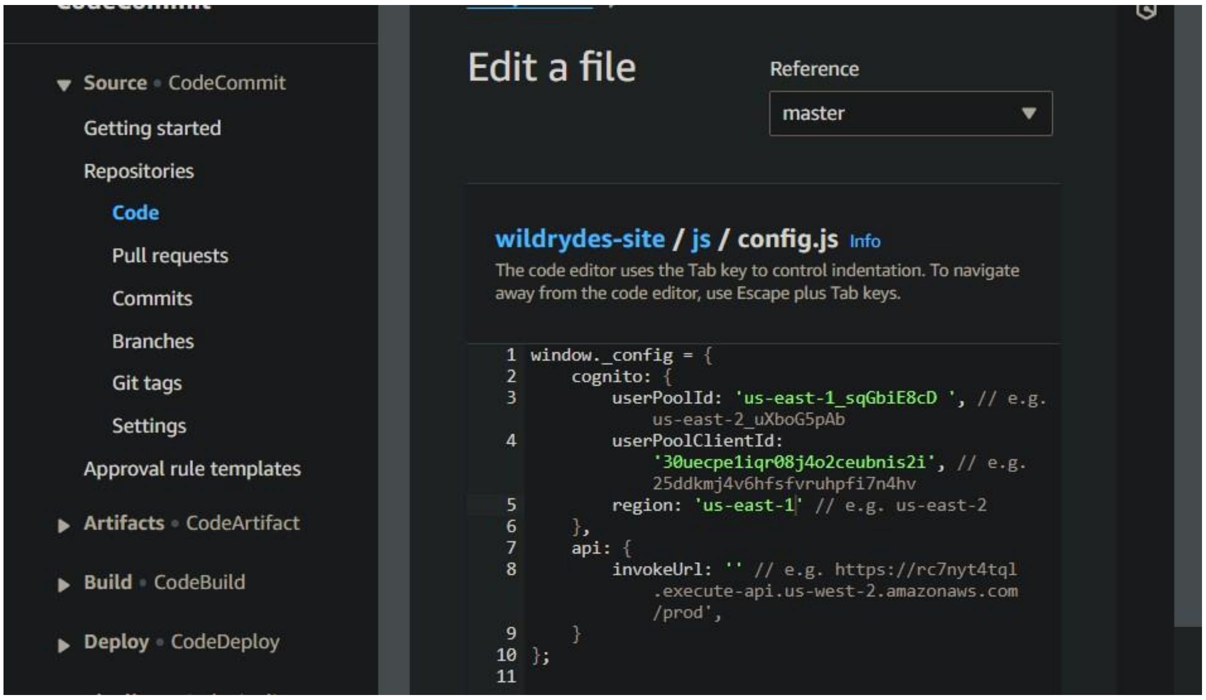
Integrate Amazon Cognito with Amazon Verified Permissions
Amazon Verified Permissions is a fine-grained authorization service for role- and attribute-based access control in apps that authenticate with Amazon Cognito. From a user's identity or access token, Verified Permissions compares their attributes to access rules for your resource. The response is an authorization decision: allow or deny. Verified Permissions can consolidate authorization for all of your apps and resources into a central policy store. Your policies are written in Cedar, an open-source language built for access control. Learn more

User pools (1)
View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.

User pool name	User pool ID	Created time	Last updated time
WildRydes	us-east-1_sqGbiE8cD	2 seconds ago	2 seconds ago

Userpool ID : us-east-1_N2*****Xp
ClientID : 3tbhnbnif********m2lgr

Now update the config file in the application file to point it to the user pool
>Go back to code commit and repo go to js file and to config .js and it by putting the Userpool ID and Client ID



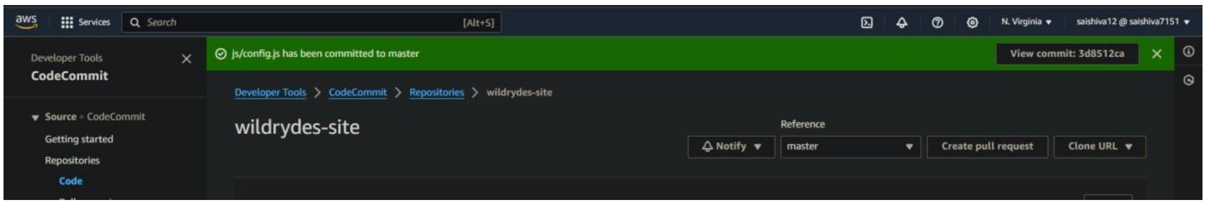
Edit a file

wildrydes-site / js / config.js Info

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
window._config = {
    cognito: {
        userPoolId: 'us-east-1_sqGbiE8cD ', // e.g. us-east-2_uXboG5pAb
        userPoolClientId: '30uecpe1iqr08j4o2ceubnis2i', // 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',
    }
};
```

We will change the url later so now commit the changes

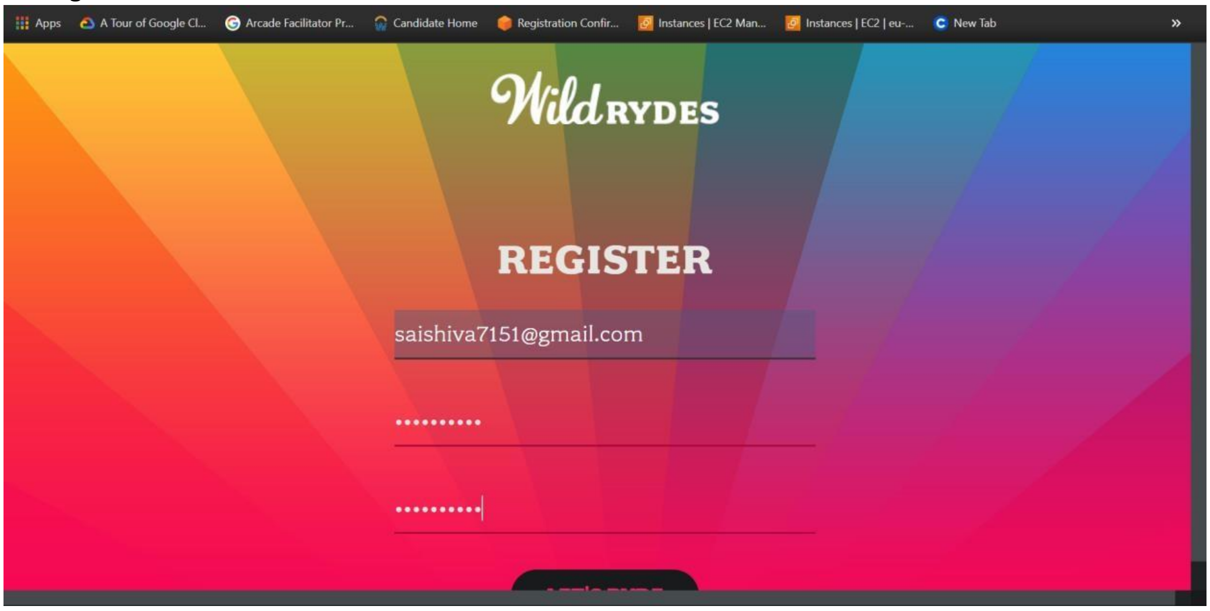


js/config.js has been committed to master

View commit: 3d8512ca

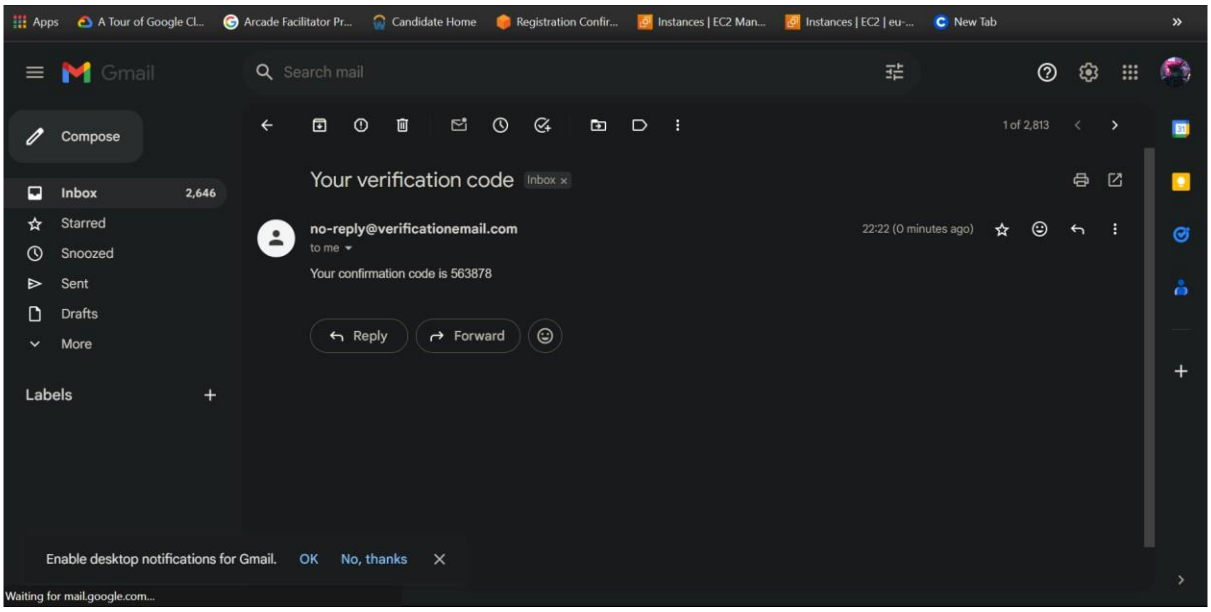
wildrydes-site

>> Registration commits success



saishiva7151@gmail.com

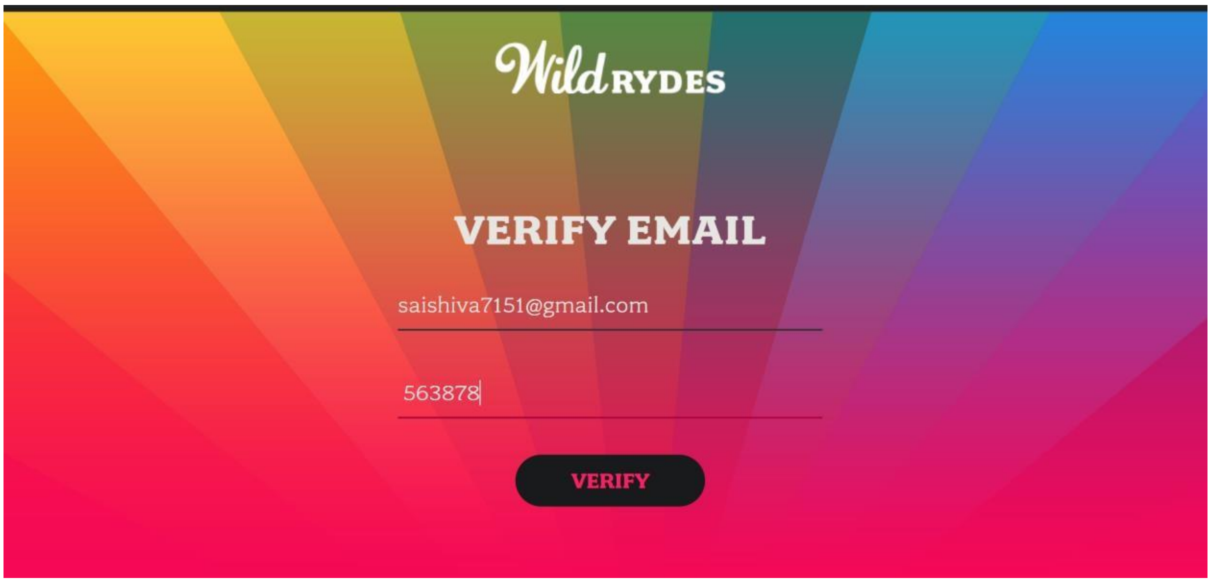
It should have sent a conformation code to our emal let us check it now


Your verification code

no-reply@verificationemail.com

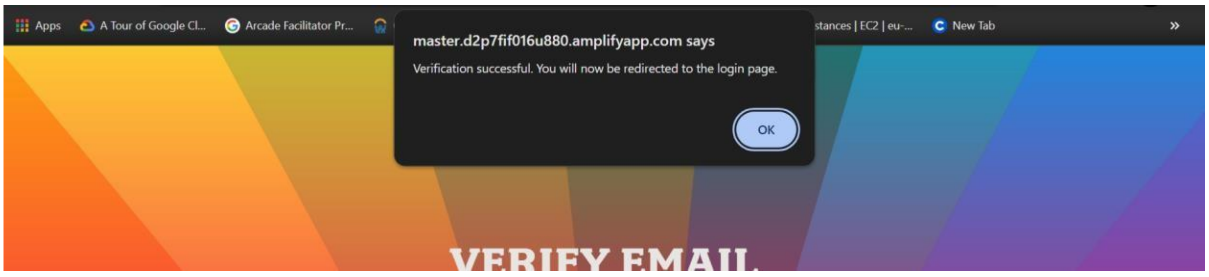
to me

Your confirmation code is 563878

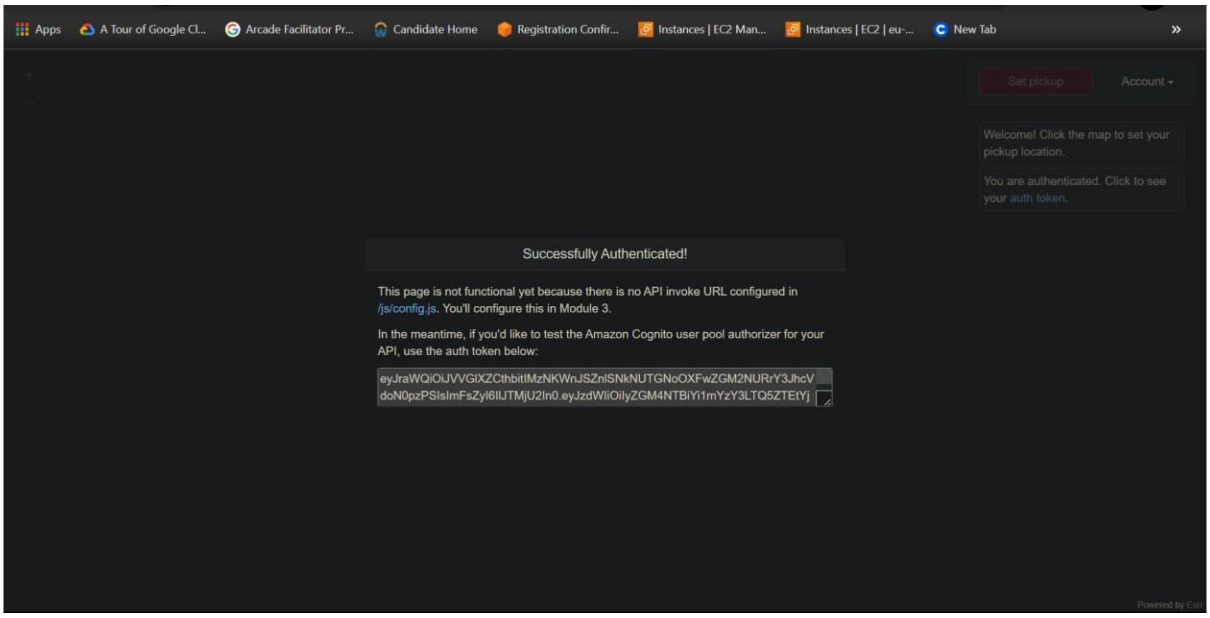


saishiva7151@gmail.com

563878


master.d2p7fif016u880.amplifyapp.com says

Verification successful. You will now be redirected to the login page.



Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Successfully Authenticated!

This page is not functional yet because there is no API invoke URL configured in /js/config.js. You'll configure this in Module 3.

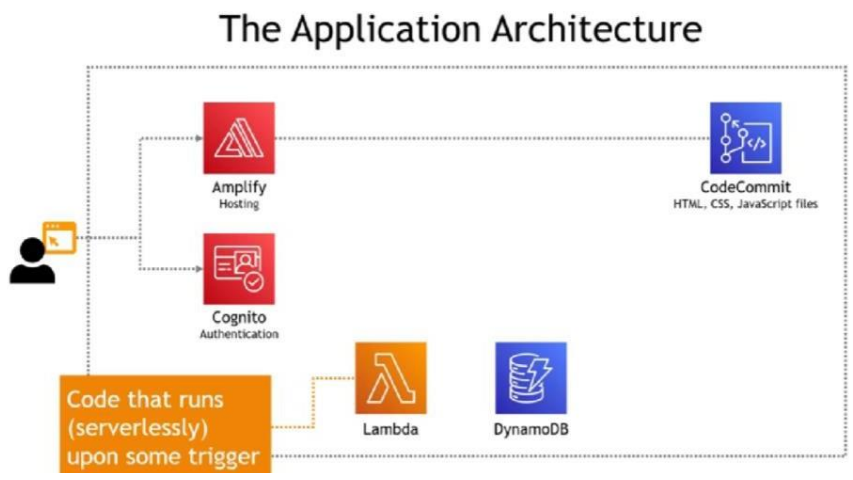
In the meantime, if you'd like to test the Amazon Cognito user pool authorizer for your API, use the auth token below:

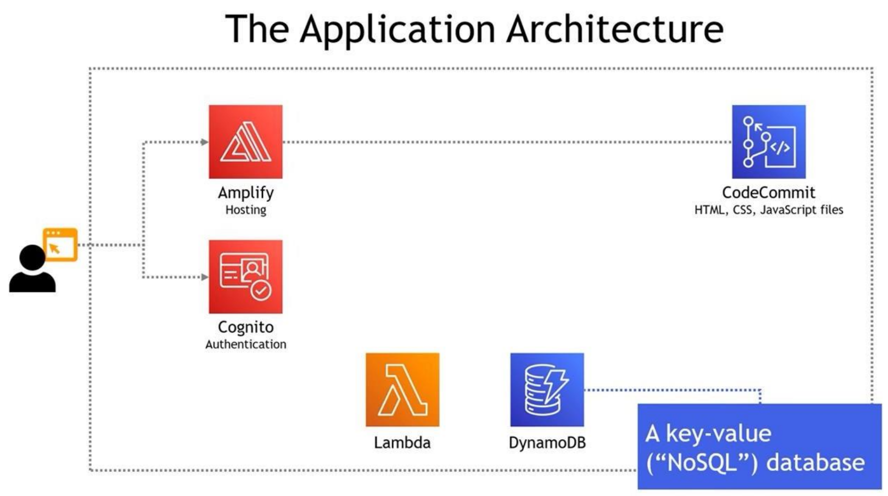
```
eyJraWQiOiJVVGlXZCthbitIMzNKWnJSZnlSNkNUTGNoOXFwZGM2NURrY3JhcV
doN0pzPSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOilyZGM4NTBiYi1mYzY3LTQ5ZTEtYj
```

Successfully authenticated and the authecated token is as given below

eyJraWQiOiJVVGlXZCthbitIMzNKWnJSZnlSNkNUTGNoOXFwZGM2NURrY3JhcVdoN0pzPSIsImFsZyI6IlJT
MjU2In0.eyJzdWIiOilyZGM4NTBiYi1mYzY3LTQ5ZTEtYjVlNC0yYzI0ZDNkYjIzNGUiLCJlbWFpbF92ZXJpZml
lZCI6dHJ1ZSwiaXNzIjoiaHR0cHM6XC9cL2NvZ25pdG8taWRwLnVzLWVhc3QtMS5hbWF6b25hd3MuY2
9tXC91cy1lYXN0LTFfTjJVc3MyclhwIiwiY29nbml0bzp1c2VybmFtZSI6InNhaXNoaXZhZhNzE1MS1hdC1nb
WFpbC5jb20iLCJvcmlnaW5fanRpIjoiNWRiZTA1MTktNjI0MC00N2NjLTkwNzktMGU1MmFkZGYzM2EwI
iwiYXVkIjoiM3RiaG5ibmlmdmRqOTMxbDNuMmoybTJsZ3IiLCJldmVudF9pZCI6ImNhZDYxNDAwLTFlM
DgtNDVmMC04ZmMzLTk1Mzg4OWMwNTQ2MSIsInRva2VuX3VzZSI6ImlkIiwiYXV0aF90aW1lIjoxNzA1
MzAwMzMwLCJleHAiOjE3MDUzMDM5MzAsImlhdCI6MTcwNTMwMDMzMCwianRpIjoiNjlyMGQwZ
mYtYjcwMS00YzZmLTkxM2MtNGNlNGY4N2FmY2ZiIiwiZW1haWwiOiJzYWlzaGl2YTcxNTFAZ21haWwu
Y29tIn0.fJlH3kzE4Fssn2hsDcKernAayDsJGAofdh4T3dvTjFf3vvPMFEE9hh8ipN9jcAEvNcW5Cy4fwnW8A
j3jJa_xXnp_ZtEPUH-
99aq_KhqDtIGPdydAi6uqp3qrr633LRV5abi8FpuTllMntAgcn9lMmyPwNKfl18VQWSSFkR0Yhux_3CSUC
RUtfT3qwPLrEamEOBSjKFWJoeZ_flInE37IWifgbUBM72bYMPYHdI8AV7QUQLtVCGINjdK8hP0kETBqVX
0PWWQI9Cxmtvrt5_qADkYu-
Dd6W9jrv6sUjKVIHPHflTreK_fSNasy0ScCUuWpuNDGU8jE7vEkYHw37ttMrg

Now let us work on ride sharing functionality ;


The Application Architecture
Amplify
Hosting
CodeCommit
HTML, CSS, JavaScript files
Cognito
Authentication
Code that runs (serverlessly) upon some trigger
Lambda
DynamoDB

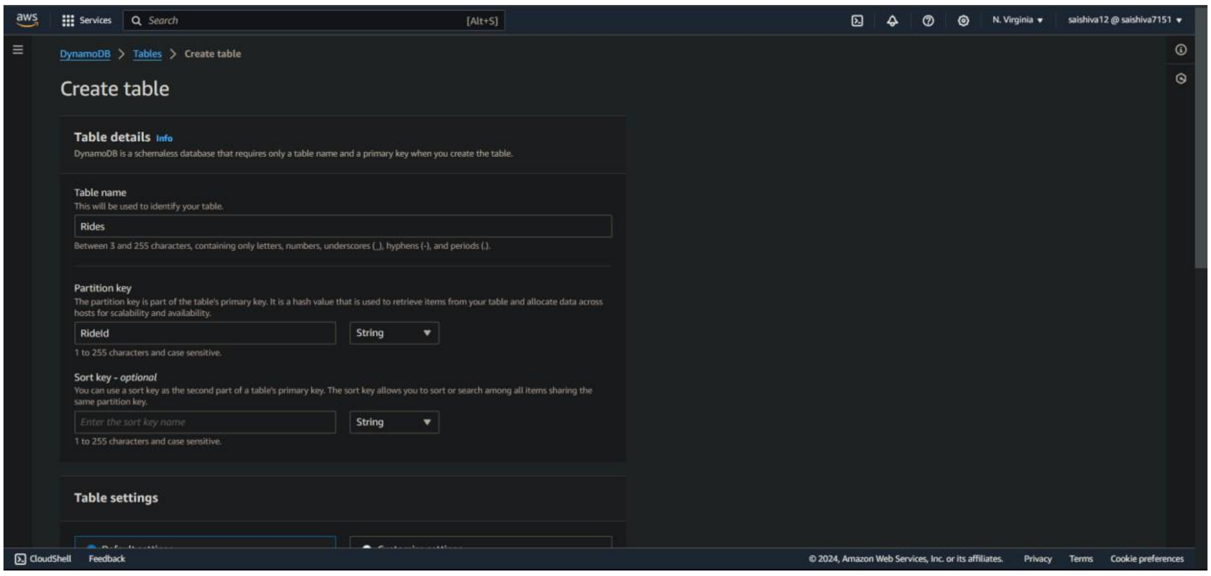

The Application Architecture
Amplify
Hosting
CodeCommit
HTML, CSS, JavaScript files
Cognito
Authentication
Lambda
DynamoDB
A key-value (“NoSQL”) database

We use lambda and DynamoDB for the enabling of a way to do ride sharing functionality

Lambda function works on triggers, In our case whenever a user requests a unicorn we use the lambda function.We use DyanmoDB as a database service which is a keyvalue or nosql database, It is a lighter weight option of Relational database.

When ever the user requests a ride that invokes the lambda function, the function selects the unicorn from the fleet and record the response in the DynamoDB table and then respond to the Frontend about the unicorn that is going to be dispatched.

First let us create a DynamoDB table



DynamoDB > Tables > Create table

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Rides

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

RideId | String

1 to 255 characters and case sensitive.

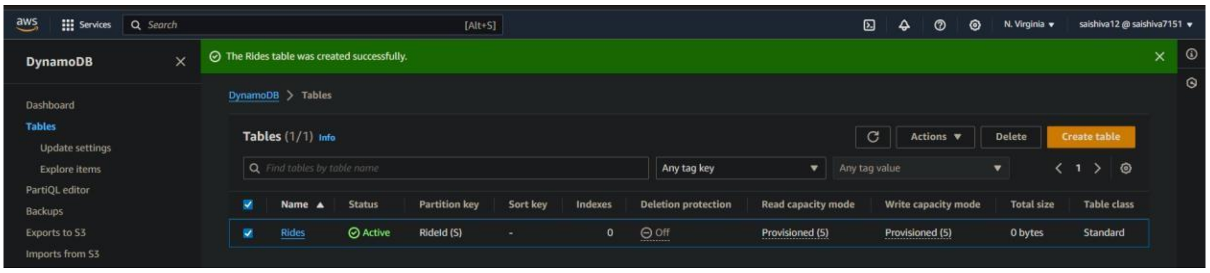
Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name | String

1 to 255 characters and case sensitive.

Table settings



The Rides table was created successfully.

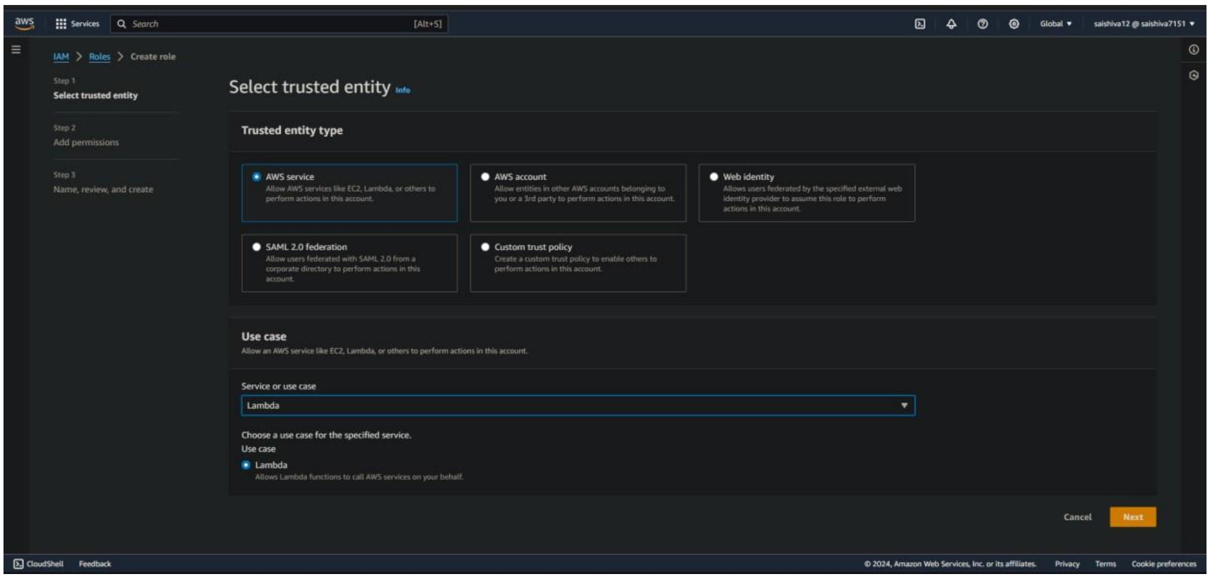
DynamoDB > Tables

Tables (1/1) Info

	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size	Table class
✓	Rides	Active	RideId (S)	-	0	Off	Provisioned (5)	Provisioned (5)	0 bytes	Standard

ARN : arn:aws:dynamodb:us-east-1:102********:table/Rides

Before Creating the lambda we need to create role in IAM function to enable Lambda to read and write to DynamoDB table.So create a role in IAM .



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity

Trusted entity type

- ● AWS service — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ○ AWS account — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ○ Web identity — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ○ SAML 2.0 federation — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ○ Custom trust policy — Create a custom trust policy to enable others to perform actions in this account.

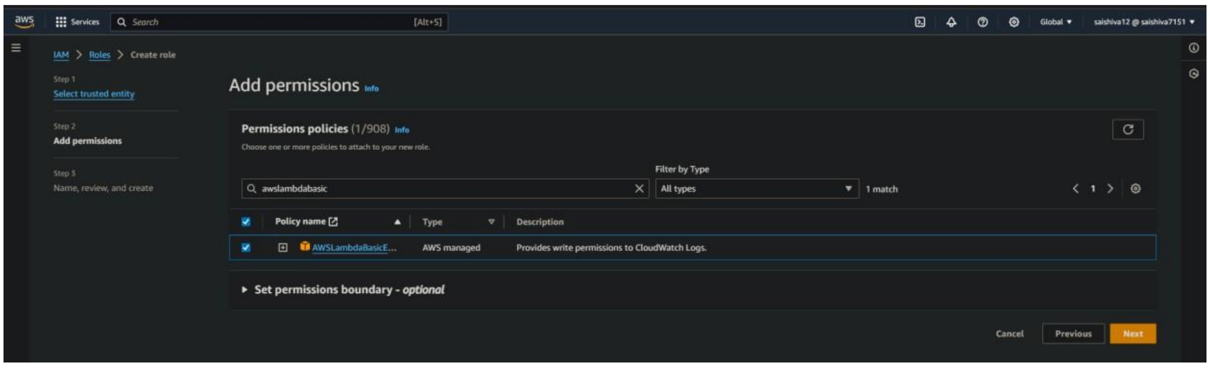
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case: Lambda

Choose a use case for the specified service.
Use case
● Lambda
Allows Lambda functions to call AWS services on your behalf.

Cancel | Next

© 2024, Amazon Web Services, Inc. or its affiliates.



IAM > Roles > Create role

Add permissions

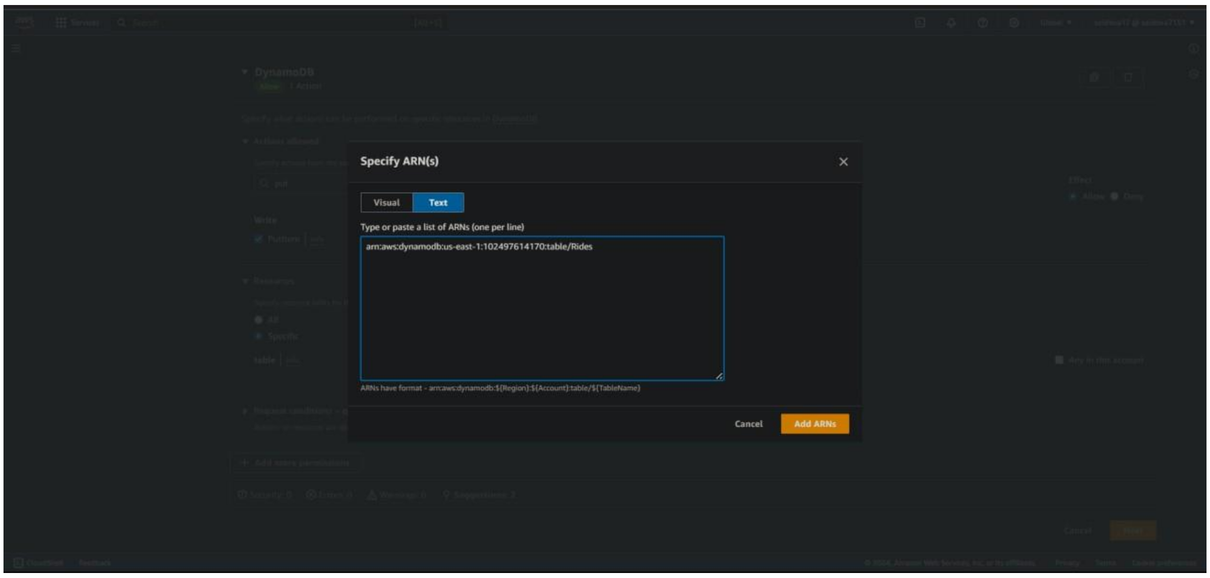
Permissions policies (1/908)
Choose one or more policies to attach to your new role.

Search: awslambdabasic | Filter by Type: All types | 1 match

	Policy name	Type	Description
✓	AWSLambdaBasicE...	AWS managed	Provides write permissions to CloudWatch Logs.

Set permissions boundary - *optional*

Cancel | Previous | Next



Specify ARN(s)

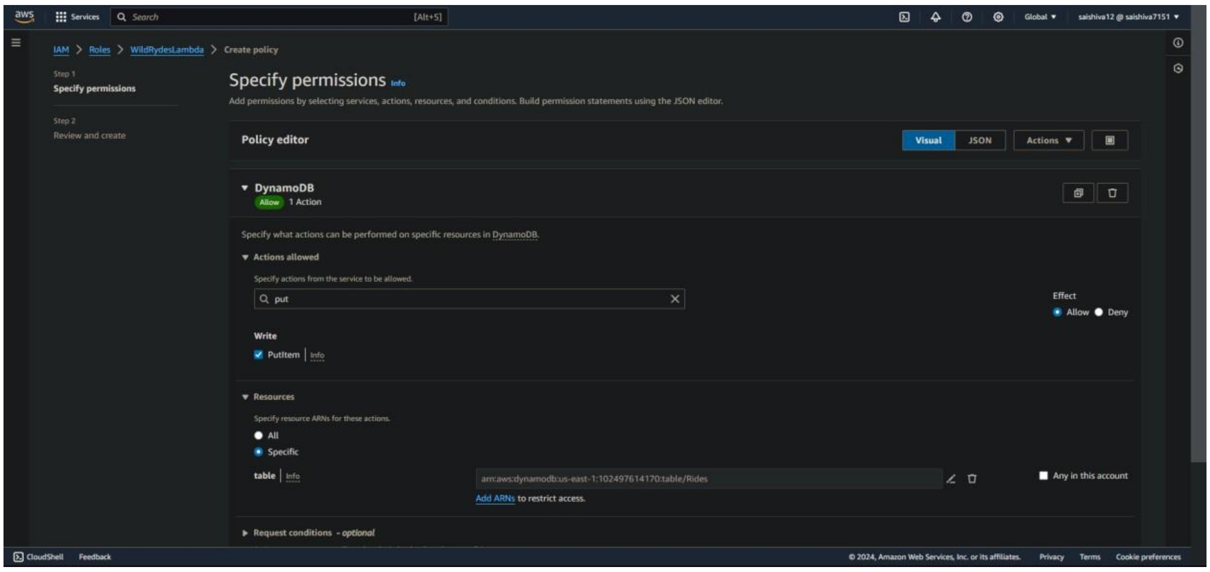
Visual | Text

Type or paste a list of ARNs (one per line)

```
arn:aws:dynamodb:us-east-1:102497614170:table/Rides
```

ARNs have format - arn:aws:dynamodb:${Region}:${Account}:table/${TableName}

Cancel | Add ARNs


IAM > Roles > WildRydesLambda > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

DynamoDB
Allow 1 Action

Specify what actions can be performed on specific resources in DynamoDB.

Actions allowed
Specify actions from the service to be allowed.
put

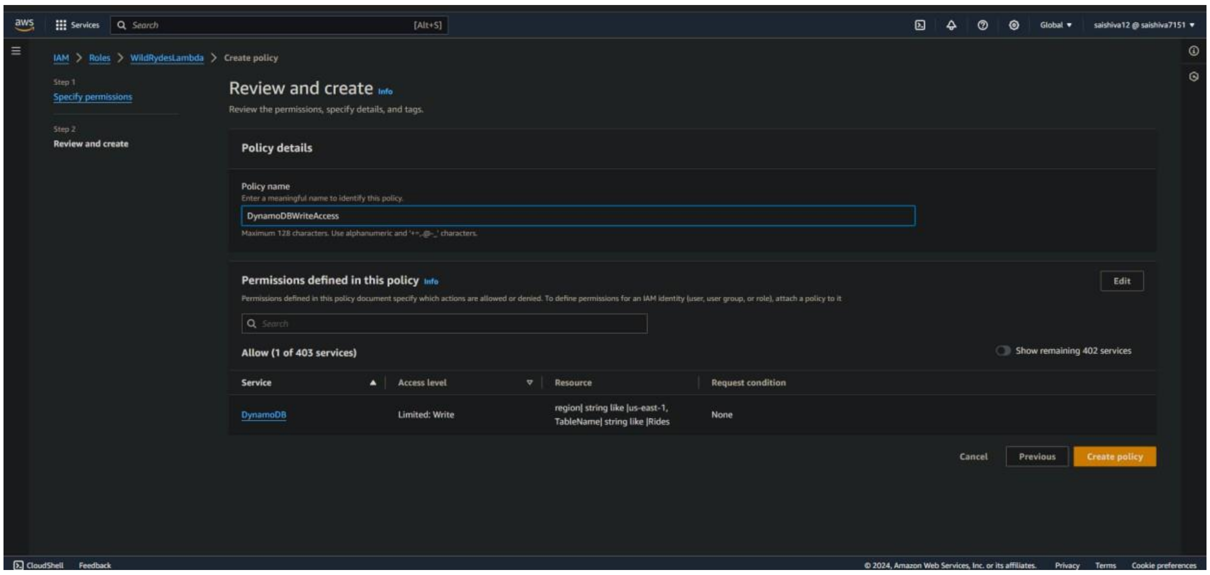
Write
✓ PutItem

Effect
● Allow ○ Deny

Resources
Specify resource ARNs for these actions.
○ All
● Specific

table: arn:aws:dynamodb:us-east-1:102497614170:table/Rides
Add ARNs to restrict access.
☐ Any in this account

Request conditions - *optional*


IAM > Roles > WildRydesLambda > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create
Review the permissions, specify details, and tags.

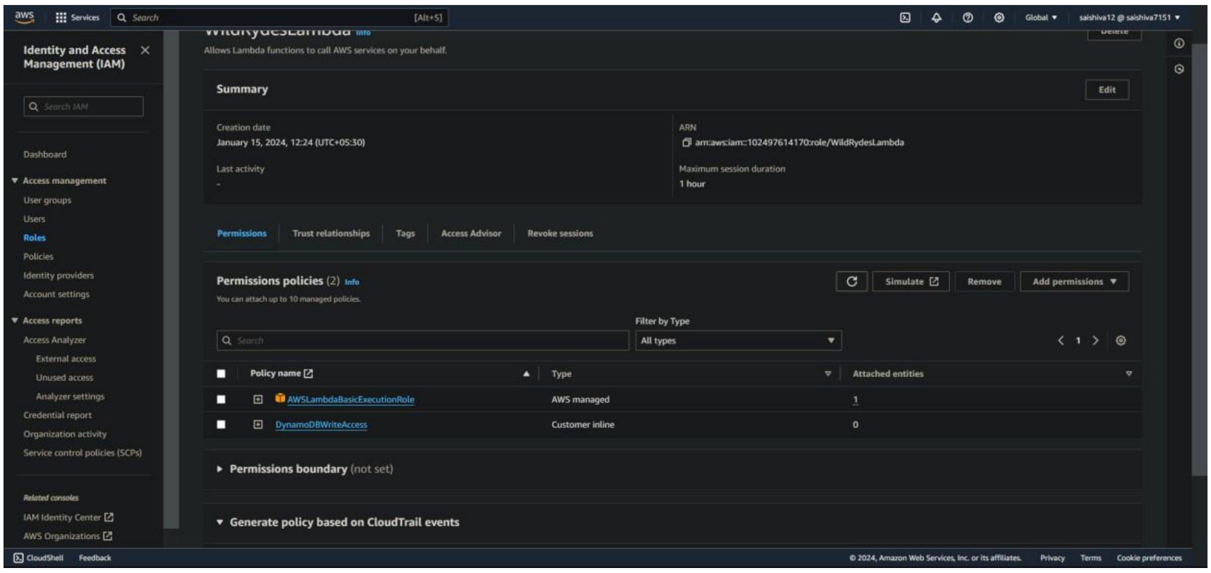
Policy details

Policy name
Enter a meaningful name to identify this policy.
DynamoDBWriteAccess
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions defined in this policy
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 403 services)
Show remaining 402 services

Service	Access level	Resource	Request condition
DynamoDB	Limited: Write	region| string like |us-east-1, TableName| string like |Rides	None


Identity and Access Management (IAM)

Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies (SCPs)
Related consoles
IAM Identity Center
AWS Organizations

WildRydesLambda
Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date: January 15, 2024, 12:24 (UTC+05:30)	ARN: arn:aws:iam::102497614170:role/WildRydesLambda
Last activity: -	Maximum session duration: 1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (2)
You can attach up to 10 managed policies.

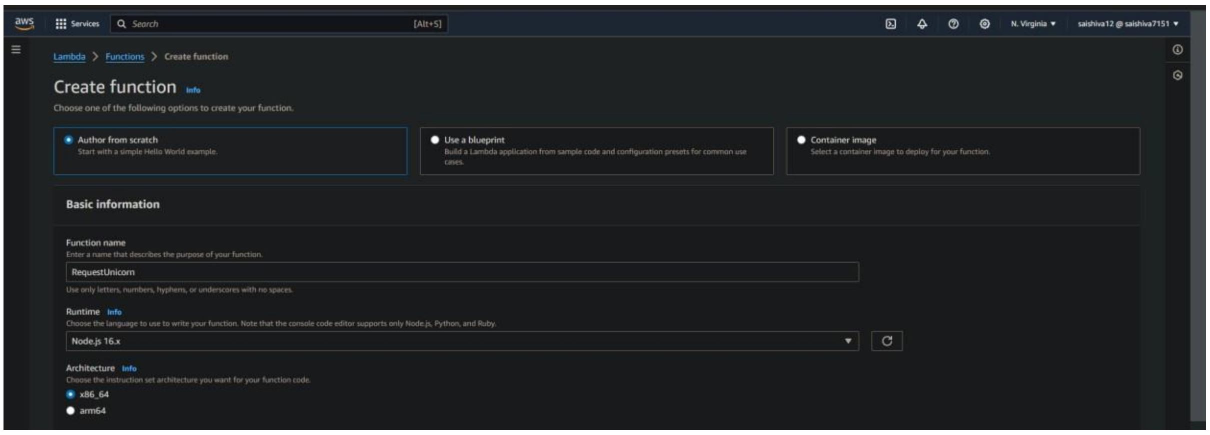
Filter by Type: All types

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole	AWS managed	1
DynamoDBWriteAccess	Customer inline	0

Permissions boundary (not set)

Generate policy based on CloudTrail events

Lambda


Lambda > Functions > Create function

Create function

Choose one of the following options to create your function.

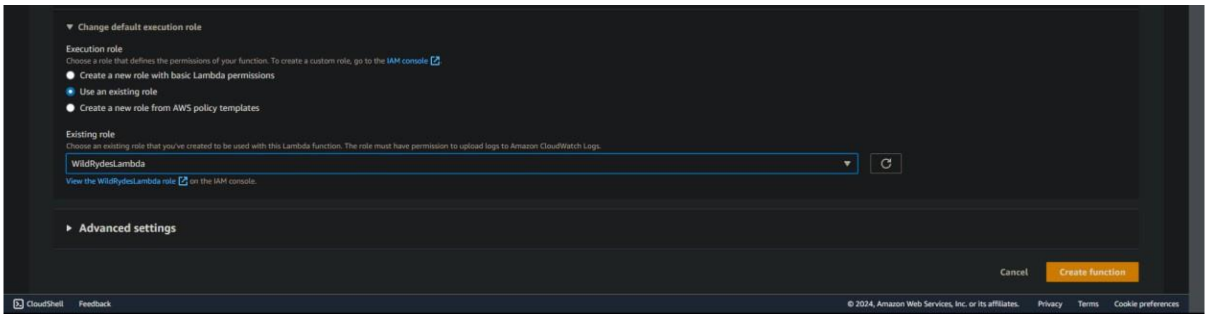
- Author from scratch — Start with a simple Hello World example.
- Use a blueprint — Build a Lambda application from sample code and configuration presets for common use cases.
- Container image — Select a container image to deploy for your function.

Basic information

Function name: RequestUnicorn

Runtime: Node.js 16.x

Architecture: x86_64 [✓] / arm64


Change default execution role

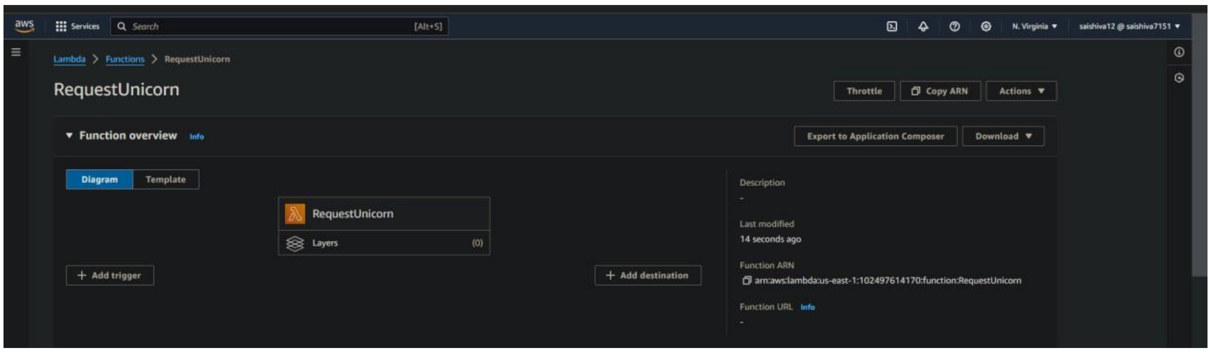
Execution role:
- Create a new role with basic Lambda permissions
- Use an existing role [✓]
- Create a new role from AWS policy templates

Existing role: WildRydesLambda

View the WildRydesLambda role on the IAM console.

Advanced settings

Cancel | Create function


Lambda > Functions > RequestUnicorn

RequestUnicorn

Function overview

RequestUnicorn — Layers (0)

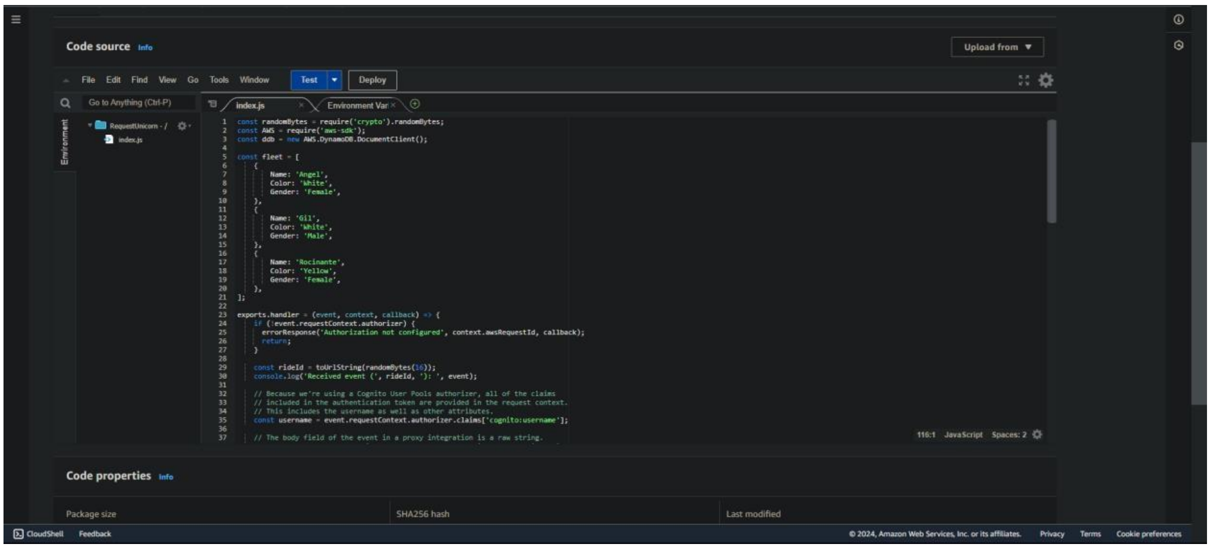
Description: -

Last modified: 14 seconds ago

Function ARN: arn:aws:lambda:us-east-1:102497614170:function:RequestUnicorn

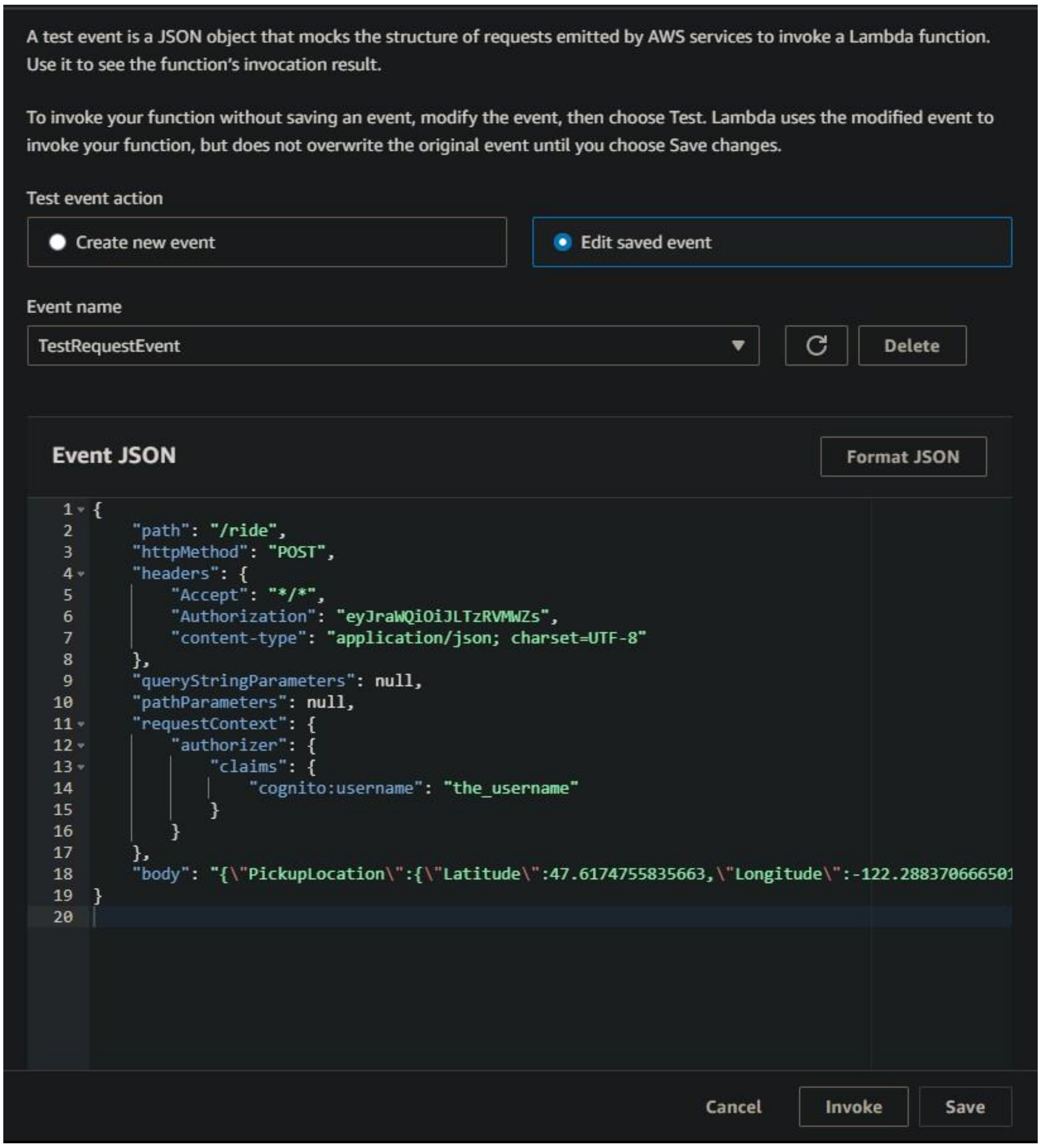
Function URL: -

Modify the code source


Code source

index.js

```
const randomBytes = require('crypto').randomBytes;
const AWS = require('aws-sdk');
const ddb = new AWS.DynamoDB.DocumentClient();

const fleet = [
    {
        Name: 'Angel',
        Color: 'White',
        Gender: 'Female',
    },
    {
        Name: 'Gil',
        Color: 'White',
        Gender: 'Male',
    },
    {
        Name: 'Rocinante',
        Color: 'Yellow',
        Gender: 'Female',
    },
];

exports.handler = (event, context, callback) => {
    if (!event.requestContext.authorizer) {
      errorResponse('Authorization not configured', context.awsRequestId, callback);
      return;
    }

    const rideId = toUrlString(randomBytes(16));
    console.log('Received event (', rideId, '): ', event);

    // Because we're using a Cognito User Pools authorizer, all of the claims
    // included in the authentication token are provided in the request context.
    // This includes the username as well as other attributes.
    const username = event.requestContext.authorizer.claims['cognito:username'];

    // The body field of the event in a proxy integration is a raw string.
```



A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

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

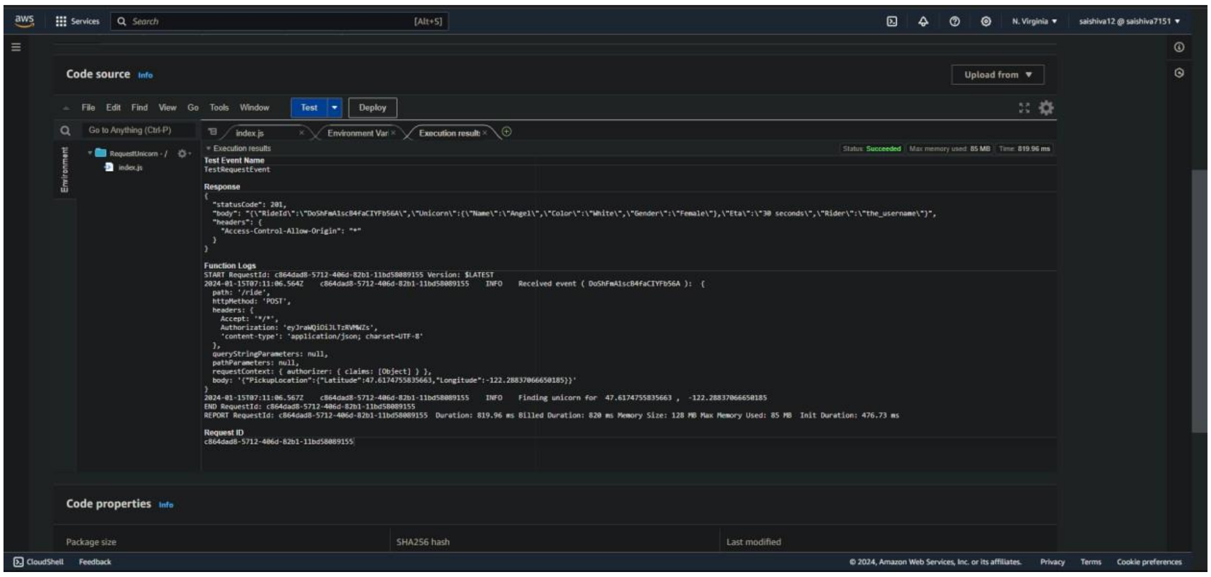
Create new event [✓ Edit saved event]

Event name

TestRequestEvent

Event JSON

```
{
    "path": "/ride",
    "httpMethod": "POST",
    "headers": {
        "Accept": "*/*",
        "Authorization": "eyJraWQiOiJLTzRVMWZs",
        "content-type": "application/json; charset=UTF-8"
    },
    "queryStringParameters": null,
    "pathParameters": null,
    "requestContext": {
        "authorizer": {
            "claims": {
                "cognito:username": "the_username"
            }
        }
    },
    "body": "{\"PickupLocation\":{\"Latitude\":47.6174755835663,\"Longitude\":-122.288370666501
}
```



Code source

Execution results — Status: Succeeded — Max memory used: 85 MB — Time: 819.96 ms

Test Event Name

TestRequestEvent

Response

```
{
  "statusCode": 201,
  "body": "{\"RideId\":\"DoShFmA1scB4faCIYFb56A\",\"Unicorn\":{\"Name\":\"Angel\",\"Color\":\"White\",\"Gender\":\"Female\"},\"Eta\":\"30 seconds\",\"Rider\":\"the_username\"}",
  "headers": {
    "Access-Control-Allow-Origin": "*"
  }
}
```

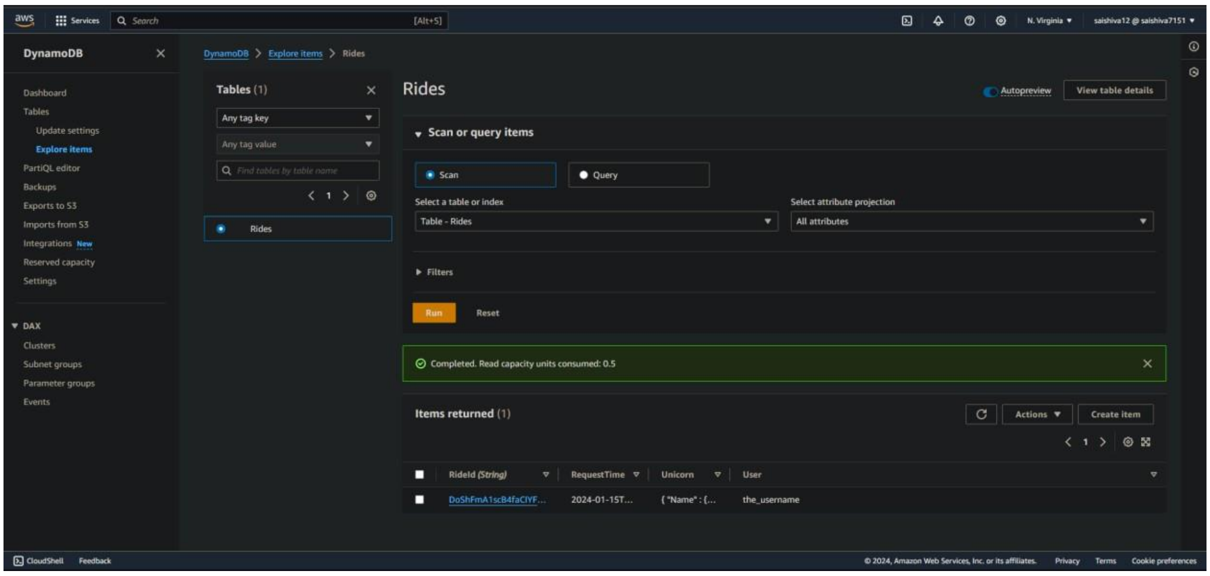
Function Logs

```
START RequestId: c864dad8-5712-406d-82b1-11bd58089155 Version: $LATEST
2024-01-15T07:11:06.564Z    c864dad8-5712-406d-82b1-11bd58089155    INFO    Received event ( DoShFmA1scB4faCIYFb56A ):  {
  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}}'
}
2024-01-15T07:11:06.567Z    c864dad8-5712-406d-82b1-11bd58089155    INFO    Finding unicorn for  47.6174755835663 ,  -122.28837066650185
END RequestId: c864dad8-5712-406d-82b1-11bd58089155
REPORT RequestId: c864dad8-5712-406d-82b1-11bd58089155  Duration: 819.96 ms Billed Duration: 820 ms Memory Size: 128 MB Max Memory Used: 85 MB  Init Duration: 476.73 ms
```

Request ID

c864dad8-5712-406d-82b1-11bd58089155

If every works fine we will get “statusCode” : 201 and we will get some items returned in DynamoDB



Rides

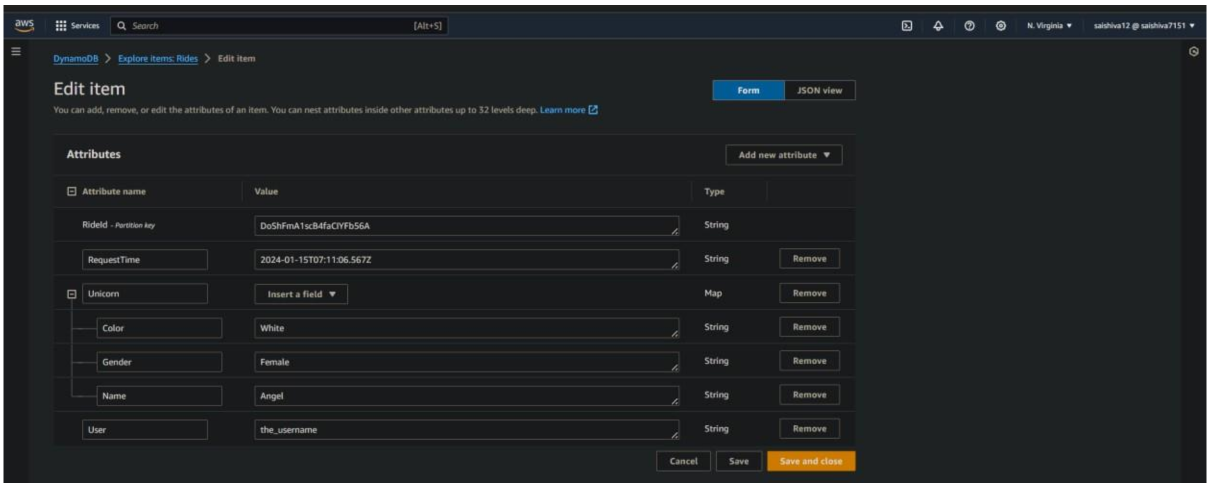
Select a table or index: Table - Rides

Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (1)

RideId (String)	RequestTime	Unicorn	User
DoShFmA1scB4faClYF...	2024-01-15T...	{ "Name" : {...	the_username



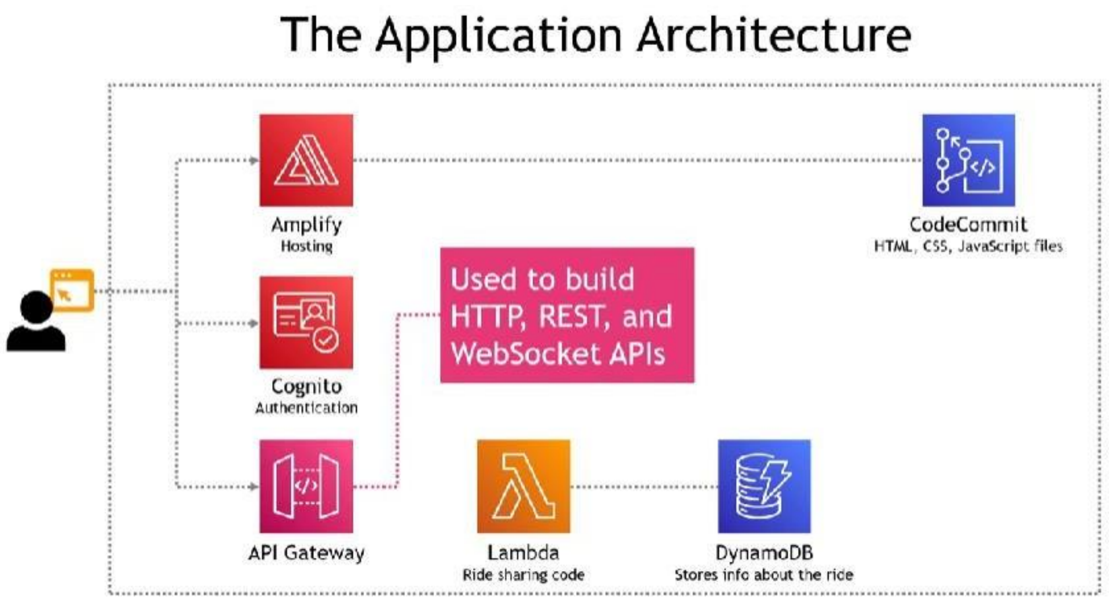
Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

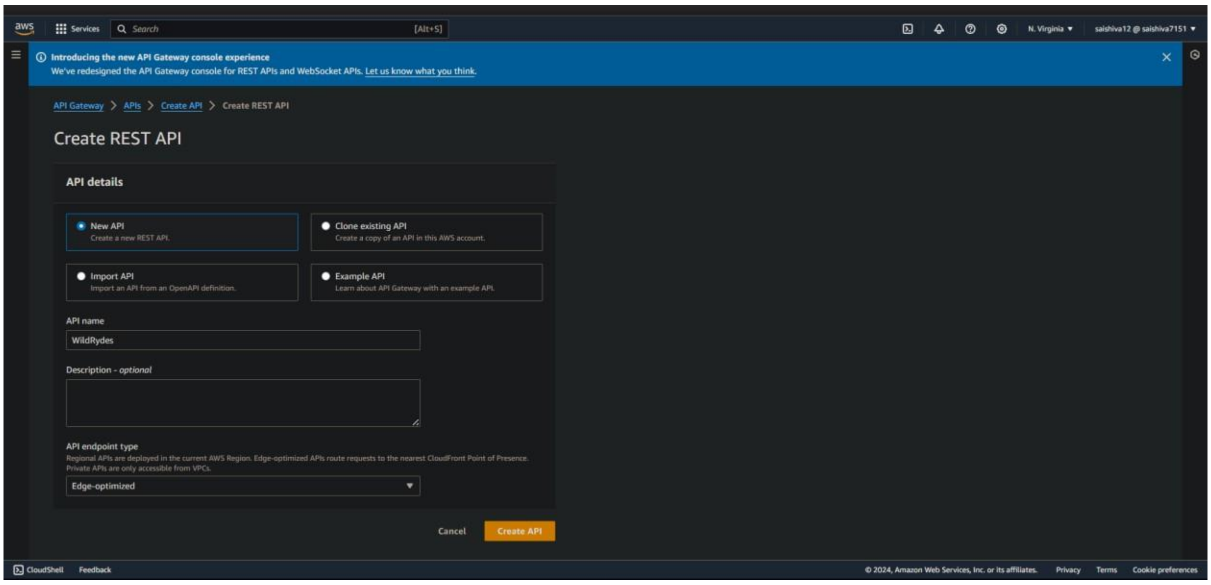
Attributes

Attribute name	Value	Type	
RideId - Partition key	DoShFmA1scB4faClYFb56A	String	
RequestTime	2024-01-15T07:11:06.567Z	String	Remove
Unicorn	Insert a field	Map	Remove
Color	White	String	Remove
Gender	Female	String	Remove
Name	Angel	String	Remove
User	the_username	String	Remove

Now for the final step of the project is to create a way to invoke ride sharing functionality. We use API Gateway to invoke the lambda function. In our case particularly REST API


The Application Architecture
Amplify
Hosting
CodeCommit
HTML, CSS, JavaScript files
Cognito
Authentication
Used to build HTTP, REST, and WebSocket APIs
API Gateway
Lambda
Ride sharing code
DynamoDB
Stores info about the ride

Making of API



Introducing the new API Gateway console experience
We've redesigned the API Gateway console for REST APIs and WebSocket APIs. Let us know what you think.

API Gateway > APIs > Create API > Create REST API

Create REST API

API details

[● New API] Create a new REST API.
○ Clone existing API — Create a copy of an API in this AWS account.
○ Import API — Import an API from an OpenAPI definition.
○ Example API — Learn about API Gateway with an example API.

API name: WildRydes

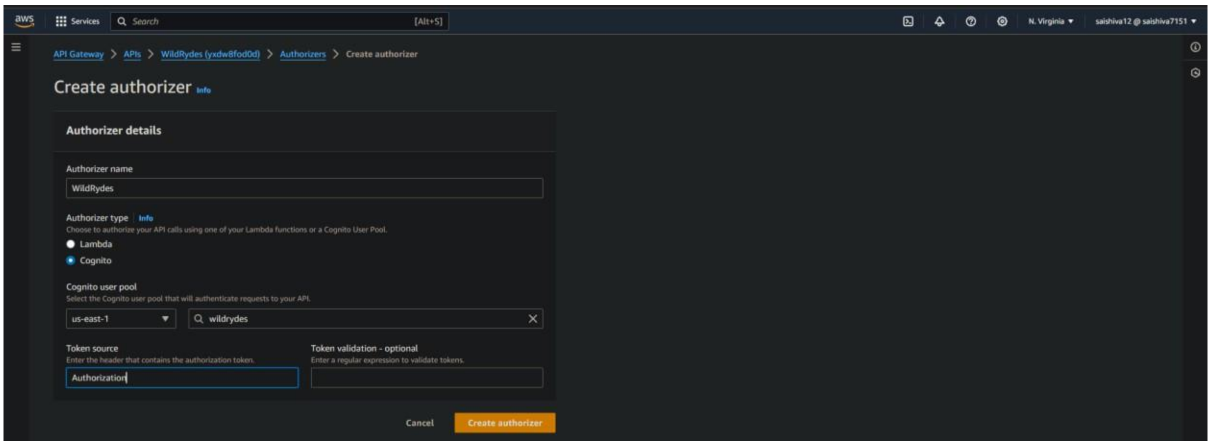
Description - optional:

API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Edge-optimized

Cancel | Create API

© 2024, Amazon Web Services, Inc. or its affiliates.

Since we are using cognito user pools we need to create an Authorizers. We need Json webtoken to authenticate calls that are returned by the cognito



API Gateway > APIs > WildRydes (yxdw8fod0d) > Authorizers > Create authorizer

Create authorizer

Authorizer details

Authorizer name: WildRydes

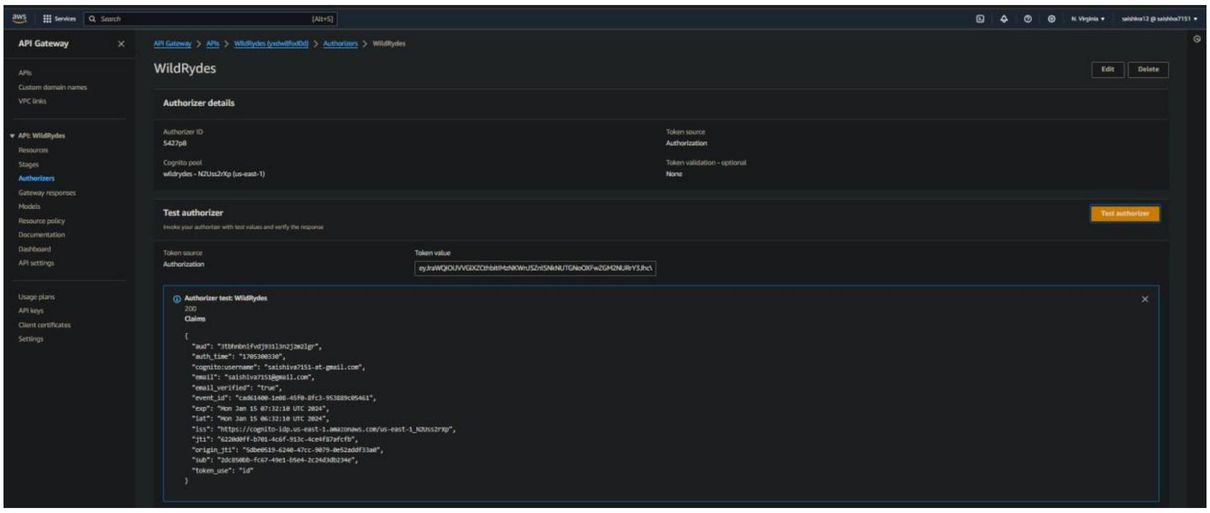
Authorizer type
Choose to authorize your API calls using one of your Lambda functions or a Cognito User Pool.
○ Lambda
[● Cognito]

Cognito user pool
Select the Cognito user pool that will authenticate requests to your API.
us-east-1 | wildrydes

Token source
Enter the header that contains the authorization token.
Authorization

Token validation - optional
Enter a regular expression to validate tokens.

Cancel | Create authorizer



API Gateway > APIs > WildRydes (yxdw8fod0d) > Authorizers > WildRydes

WildRydes

Authorizer details

Authorizer ID: 5427p8
Cognito pool: wildrydes - N2Uss2rXp (us-east-1)
Token source: Authorization
Token validation - optional: None

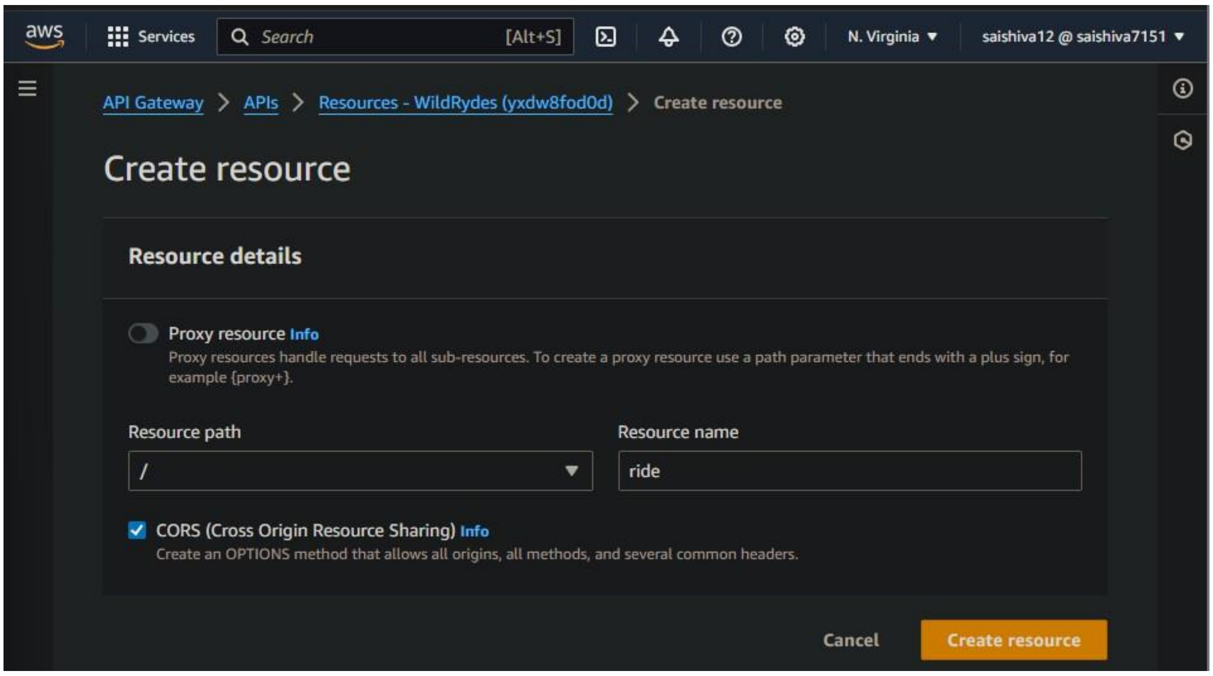
Test authorizer
Invoke your authorizer with test values and verify the response

Token source: Authorization
Token value: eyJraWQiOUVVGlXZCthbitIMzNKWnJSZnlSNkNUTGNoOXFwZGM2NURrY3Jhc\

Authorizer test: WildRydes
200
Claims

```
{
  "aud": "3tbhnbn1fvdj93113n2j2m2lgr",
  "auth_time": "1705300330",
  "cognito:username": "saishiva7151-at-gmail.com",
  "email": "saishiva7151@gmail.com",
  "email_verified": "true",
  "event_id": "cad61400-1e08-45f0-8fc3-953889c05461",
  "exp": "Mon Jan 15 07:32:10 UTC 2024",
  "iat": "Mon Jan 15 06:32:10 UTC 2024",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_N2Uss2rXp",
  "jti": "6220d0ff-b701-4c6f-913c-4ce4f87afcfb",
  "origin_jti": "5dbe0519-6240-47cc-9079-0e52addf33a0",
  "sub": "2dc850bb-fc67-49e1-b5e4-2c24d3db234e",
  "token_use": "id"
}
```

Now we need to create a resource in API so that we can hook lambda function to it.


API Gateway > APIs > Resources - WildRydes (yxdw8fod0d) > Create resource

Create resource

Resource details

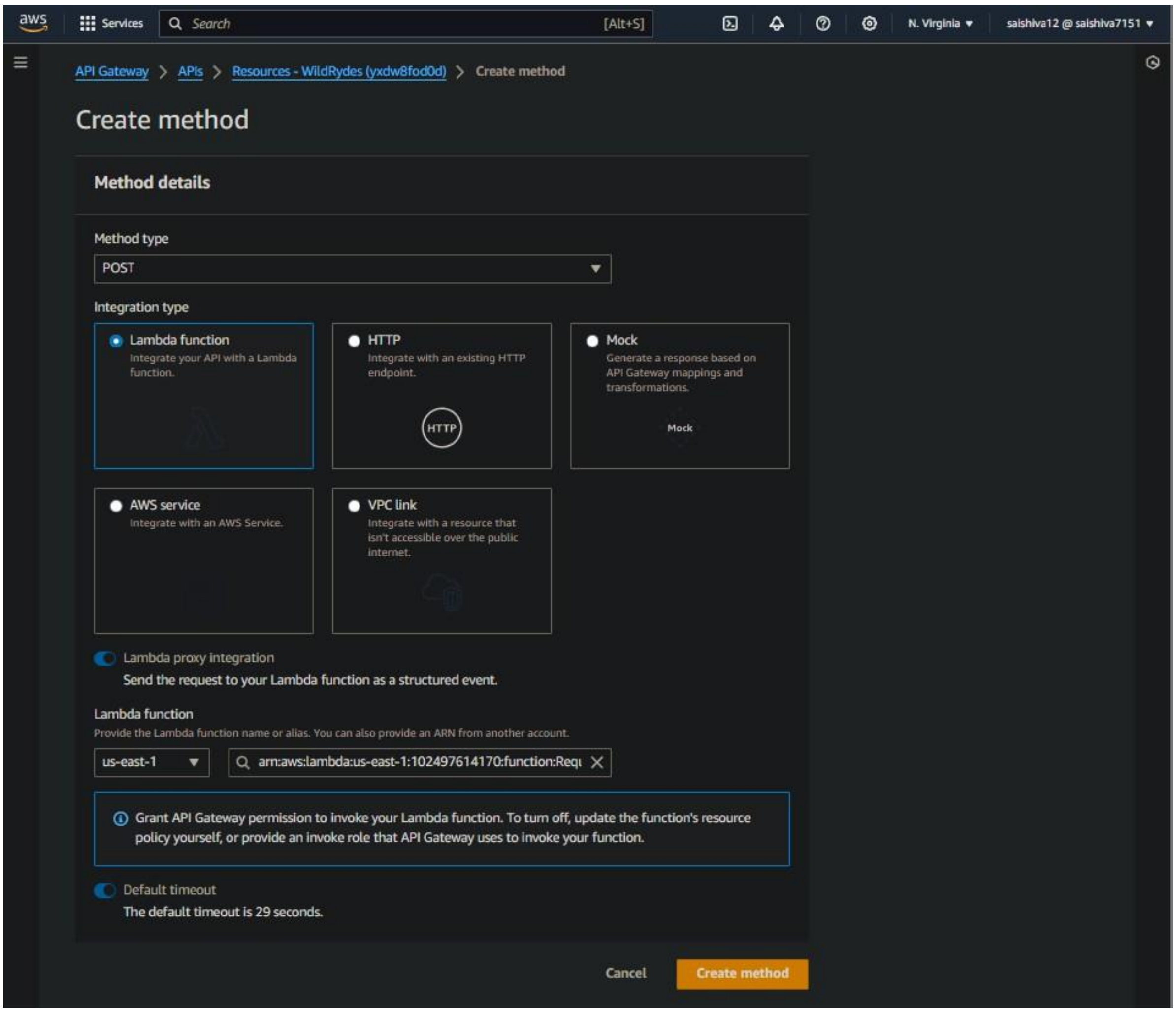
Proxy resource Info
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path: /

Resource name: ride

CORS (Cross Origin Resource Sharing) Info [✓]
Create an OPTIONS method that allows all origins, all methods, and several common headers.

Now Create method in /ride


API Gateway > APIs > Resources - WildRydes (yxdw8fod0d) > Create method

Create method

Method details

Method type: POST

Integration type

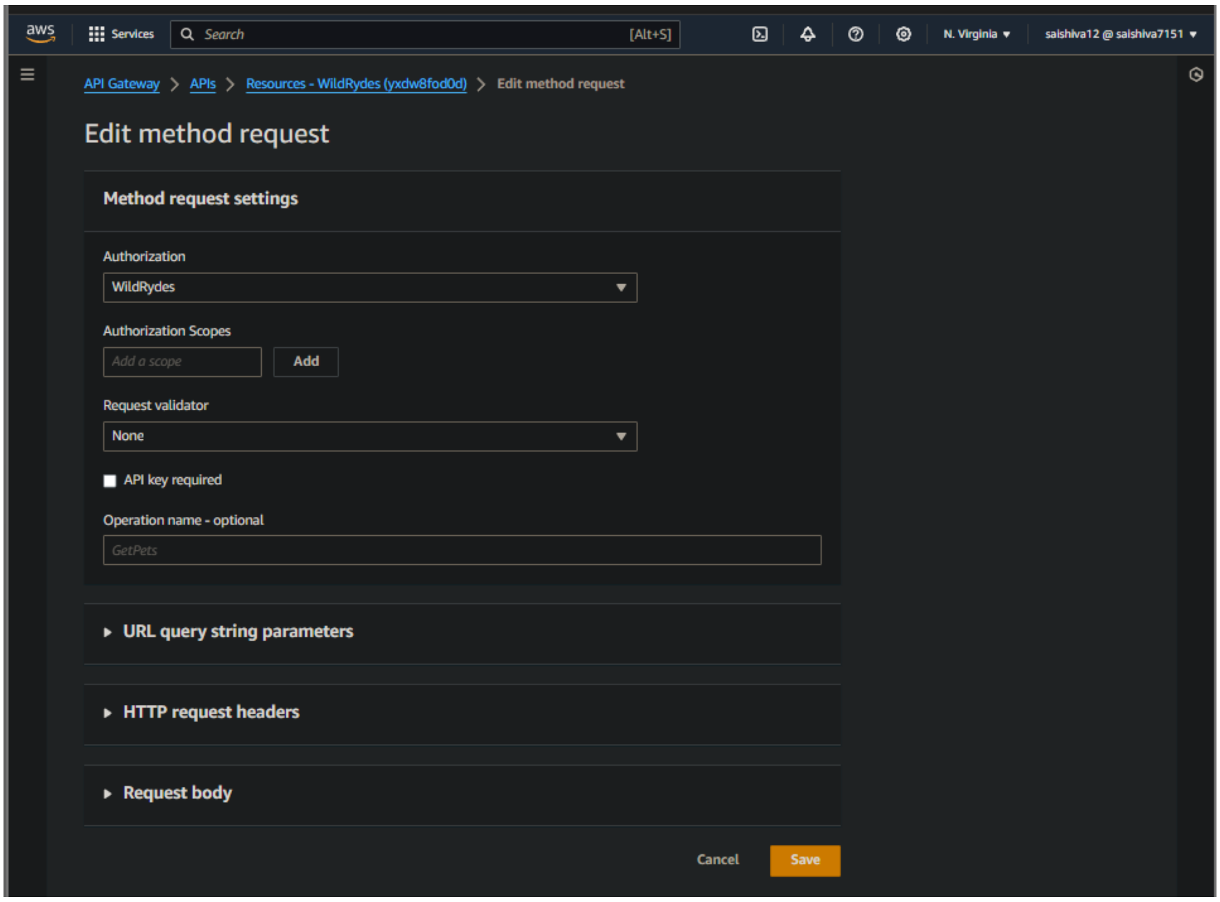
[✓ Lambda function] Integrate your API with a Lambda function.
HTTP — Integrate with an existing HTTP endpoint.
Mock — Generate a response based on API Gateway mappings and transformations.
AWS service — Integrate with an AWS Service.
VPC link — Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
us-east-1 | arn:aws:lambda:us-east-1:102497614170:function:Requ

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

Default timeout
The default timeout is 29 seconds.



API Gateway > APIs > Resources - WildRydes (yxdw8fod0d) > Edit method request

Edit method request

Method request settings

Authorization: WildRydes

Authorization Scopes: Add a scope

Request validator: None

API key required

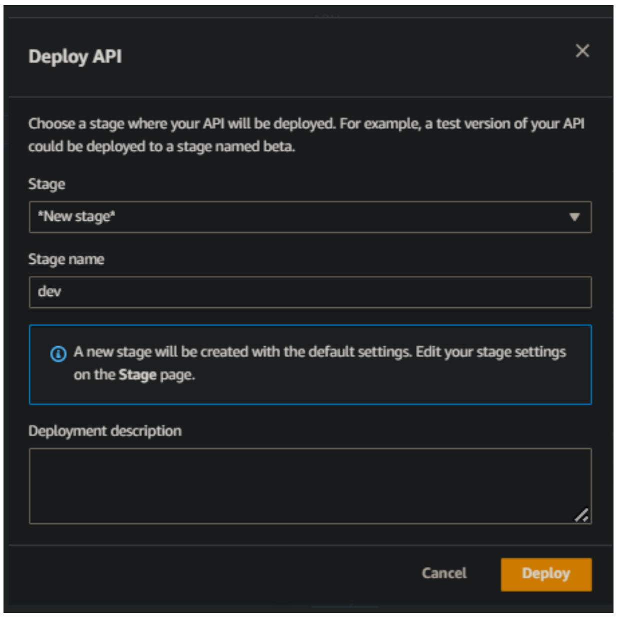
Operation name - optional: GetPets

URL query string parameters

HTTP request headers

Request body

Now Deploy the API



Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

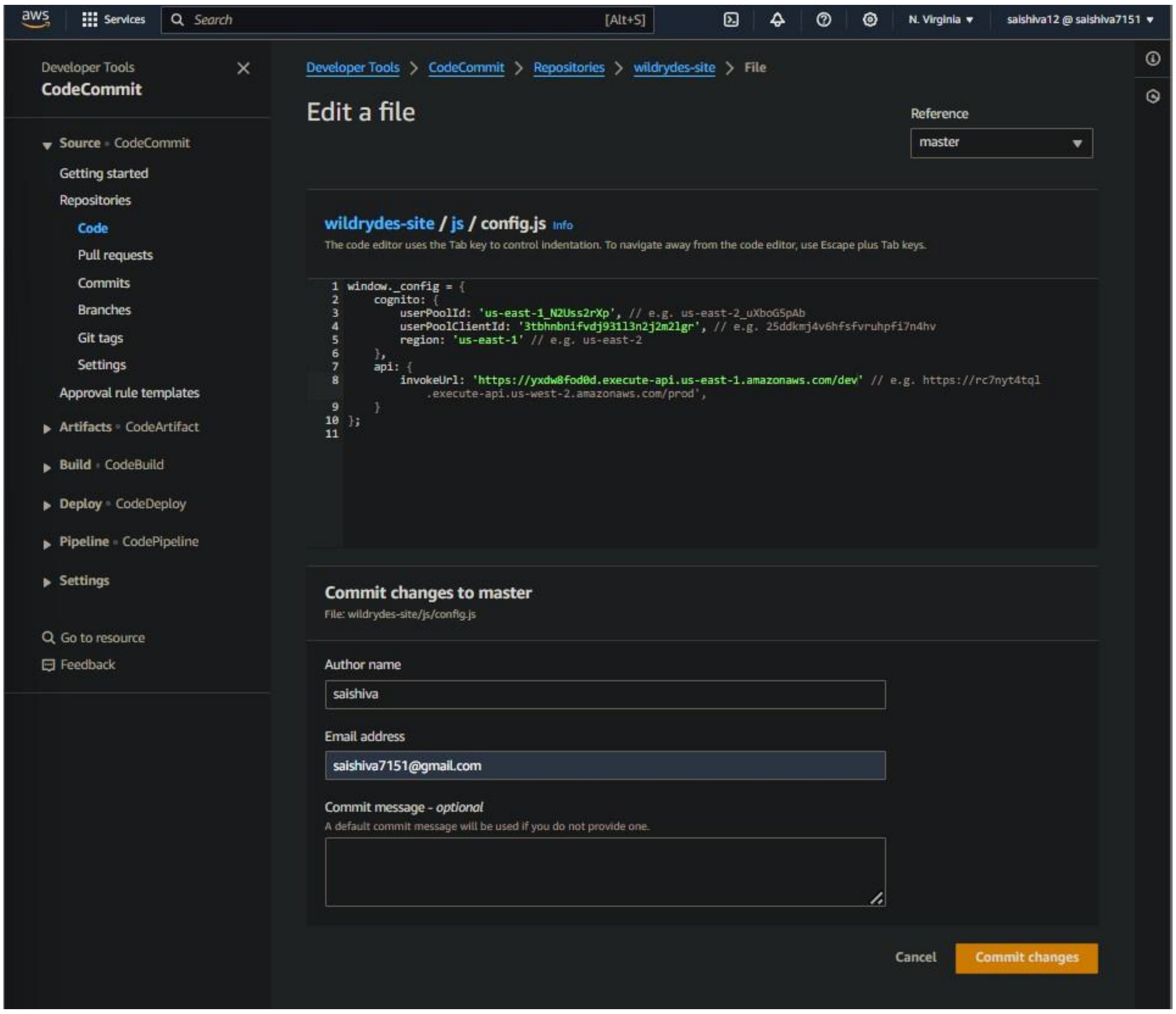
Stage: *New stage*

Stage name: dev

A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description

Invoke url : https://yxdw8fod0d.execute-api.us-east-1.amazonaws.com/dev



Edit a file

wildrydes-site / js / config.js Info

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
window._config = {
    cognito: {
        userPoolId: 'us-east-1_N2Uss2rXp', // e.g. us-east-2_uXboG5pAb
        userPoolClientId: '3tbhnbnifvdj93113n2j2m2lgr', // e.g. 25ddkmj4v6hfsfvruhpfi7n4hv
        region: 'us-east-1' // e.g. us-east-2
    },
    api: {
        invokeUrl: 'https://yxdw8fod0d.execute-api.us-east-1.amazonaws.com/dev' // e.g. https://rc7nyt4tql
            .execute-api.us-west-2.amazonaws.com/prod',
    }
};
```

Commit changes to master

File: wildrydes-site/js/config.js

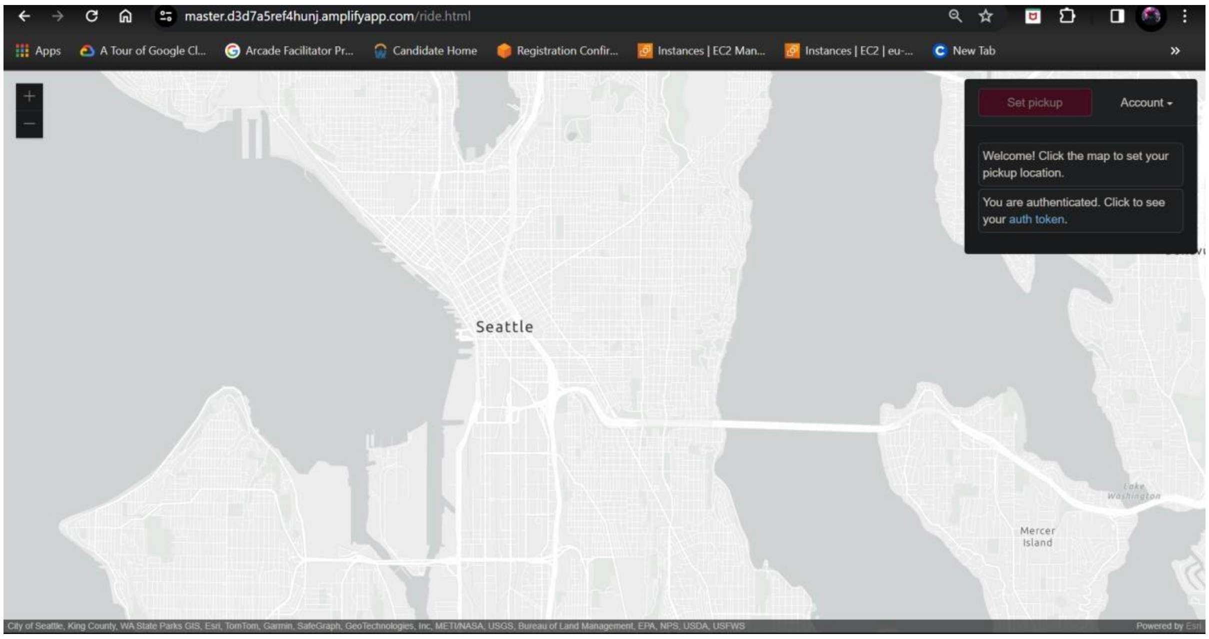
Author name: saishiva

Email address: saishiva7151@gmail.com

Commit message - *optional*

A default commit message will be used if you do not provide one.

Now reload the wildrydes app page and you will the website functioning

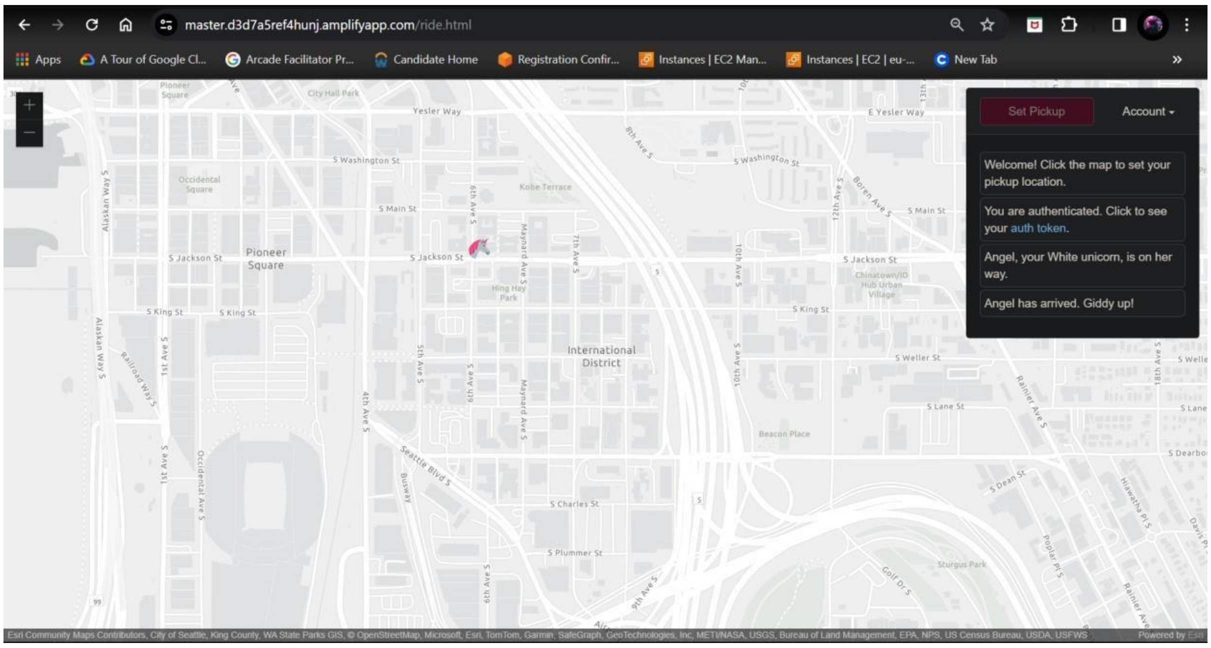


master.d3d7a5ref4hunj.amplifyapp.com/ride.html
Apps
A Tour of Google Cl...
Arcade Facilitator Pr...
Candidate Home
Registration Confir...
Instances | EC2 Man...
Instances | EC2 | eu-...
New Tab
Set pickup
Account ▾
Seattle
Mercer Island
Lake Washington
City of Seattle, King County, WA State Parks GIS, Esri, TomTom, Garmin, SafeGraph, GeoTechnologies, Inc, METI/NASA, USGS, Bureau of Land Management, EPA, NPS, USDA, USFWS
Powered by Esri

Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Note: Make sure you are logged into Arcgis account.


master.d3d7a5ref4hunj.amplifyapp.com/ride.html

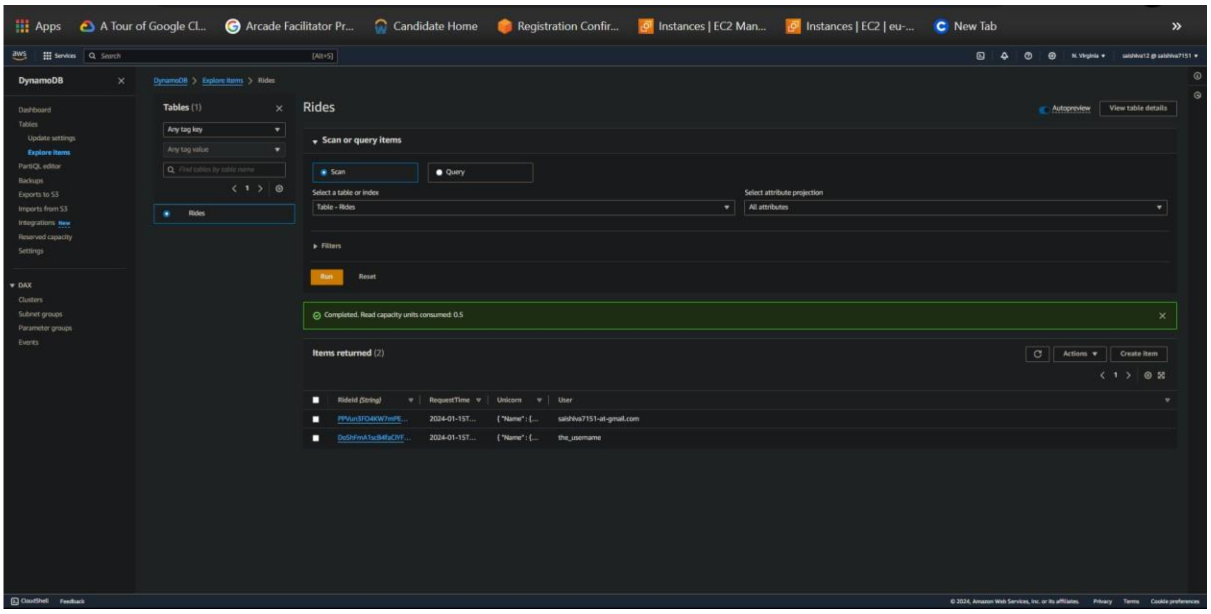
Welcome! Click the map to set your pickup location.

You are authenticated. Click to see your auth token.

Angel, your White unicorn, is on her way.

Angel has arrived. Giddy up!

We got the unicorn to the requested place. Hence it worked, Let us also check in DynamoDB if we got additional item.


DynamoDB > Explore items > Rides

Rides

Select a table or index: Table - Rides

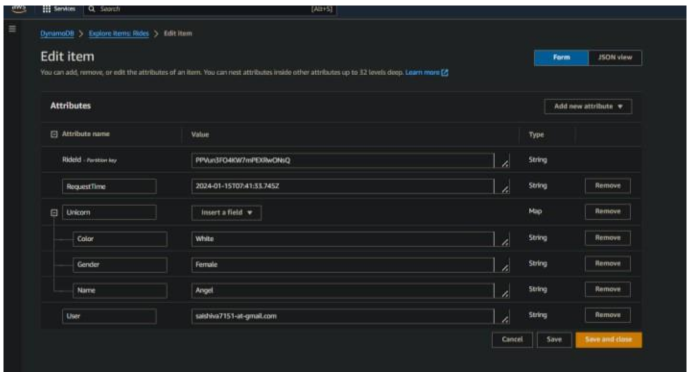
Select attribute projection: All attributes

Completed. Read capacity units consumed: 0.5

Items returned (2)

RideId (String)	RequestTime	Unicorn	User
PPVun3FO4KW7mPE...	2024-01-15T...	{ "Name" : {...	saishiva7151-at-gmail.com
DoShFmA1scB4faCIYF...	2024-01-15T...	{ "Name" : {...	the_username

We got the additional item


DynamoDB > Explore items: Rides > Edit item

Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more

Attributes

Attribute name	Value	Type
RideId - Partition key	PPVun3FO4KW7mPEXRwONsQ	String
RequestTime	2024-01-15T07:41:33.745Z	String
Unicorn		Map
Color	White	String
Gender	Female	String
Name	Angel	String
User	saishiva7151-at-gmail.com	String

Hence, The web application is successfully running