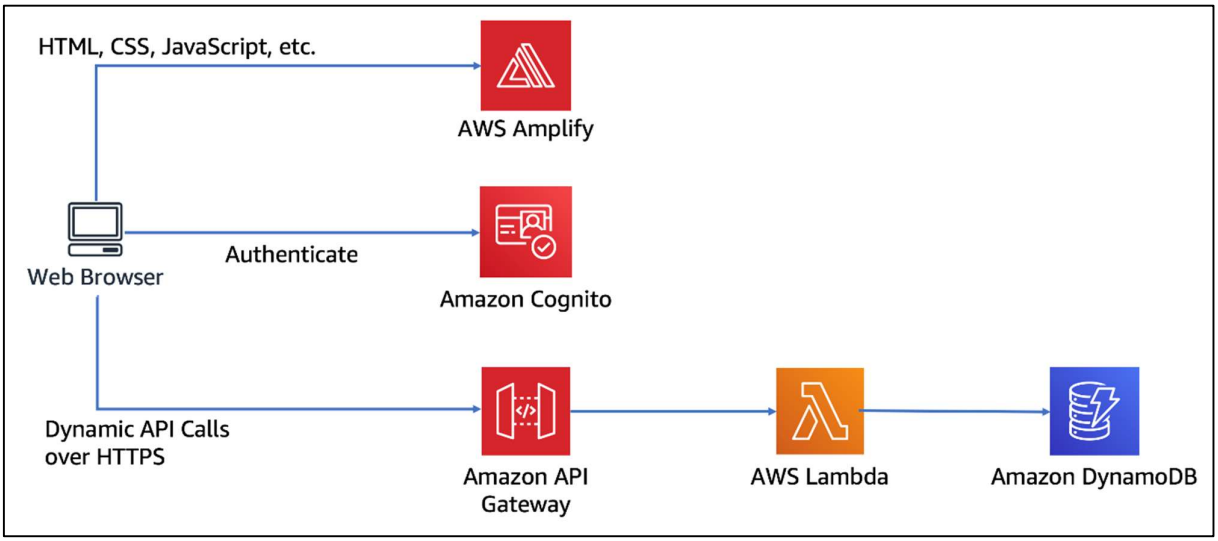
SETUP GUIDE

WildRydes is a simple web application that enables users to request unicorn rides. The application presents users with an HTML based user interface for indicating the location where they would like to be picked up and will interface on the backend with a RESTful web service to submit the request and dispatch a nearby unicorn. The application will also provide facilities for users to register with the service and log in before requesting rides.

ARCHITECTURE DIAGRAM

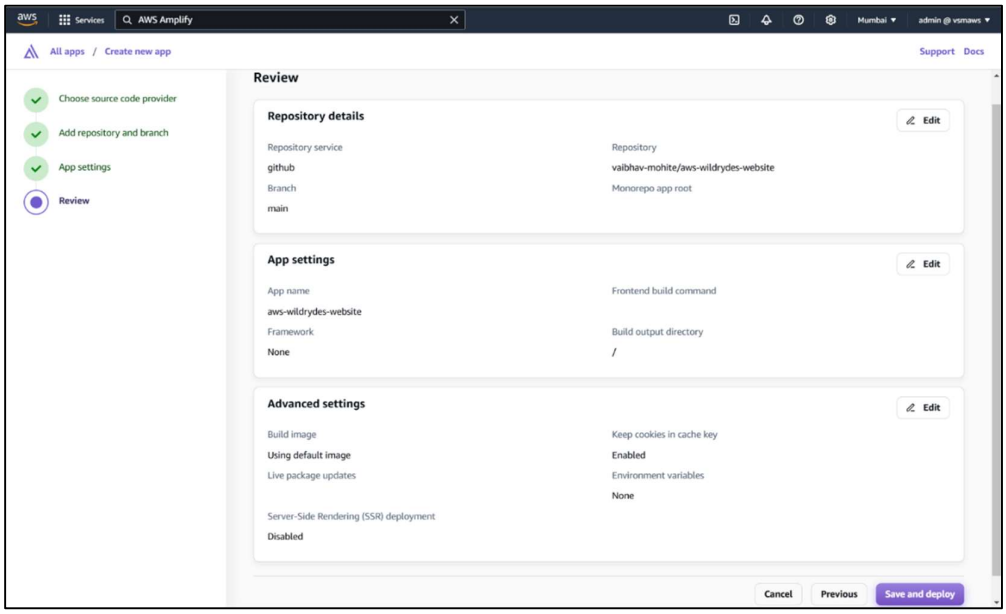


GIT REPOSITORY

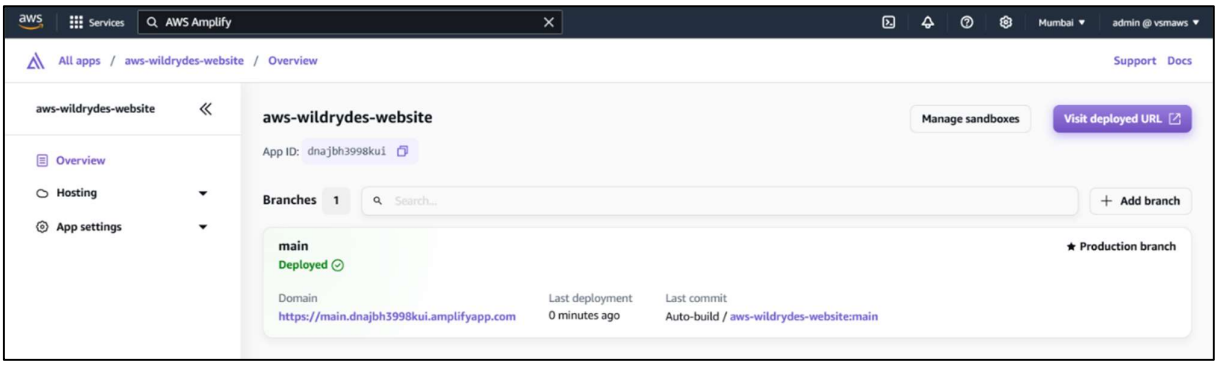
1. The repository can be found at https://github.com/vaibhav-mohite/aws-wildrydes-website

AMPLIFY

1. Launch the Amplify Console
2. Click on Deploy an App
3. Select GitHub as Git provider



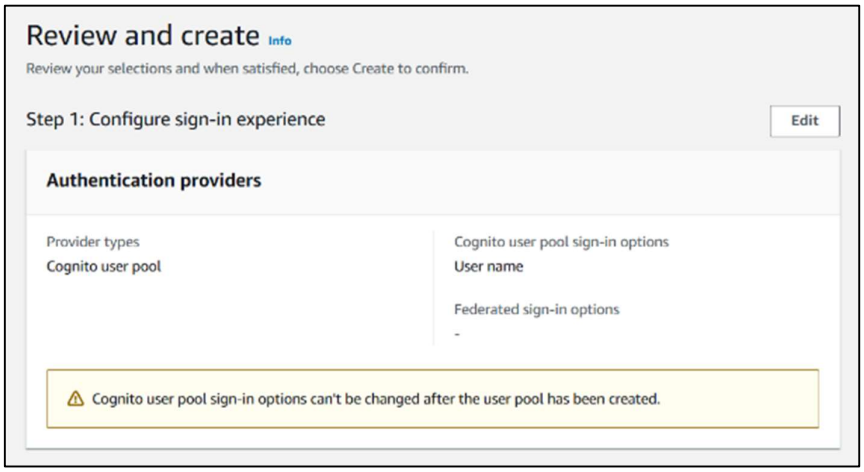
4. Connect your GitHub Account to AWS. Select the GitHub Repository.
5. Give App name as per your choice and click Save and Deploy
6. Application will be Deployed.



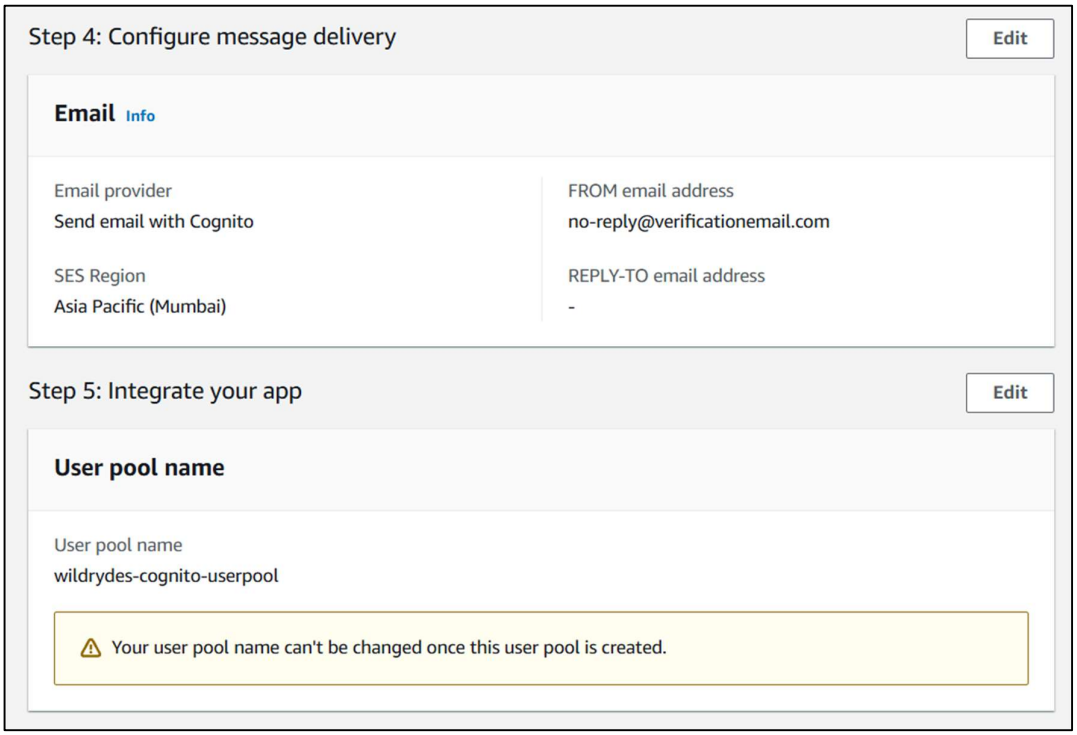
7. Click on Domain to open the website.

COGNITO

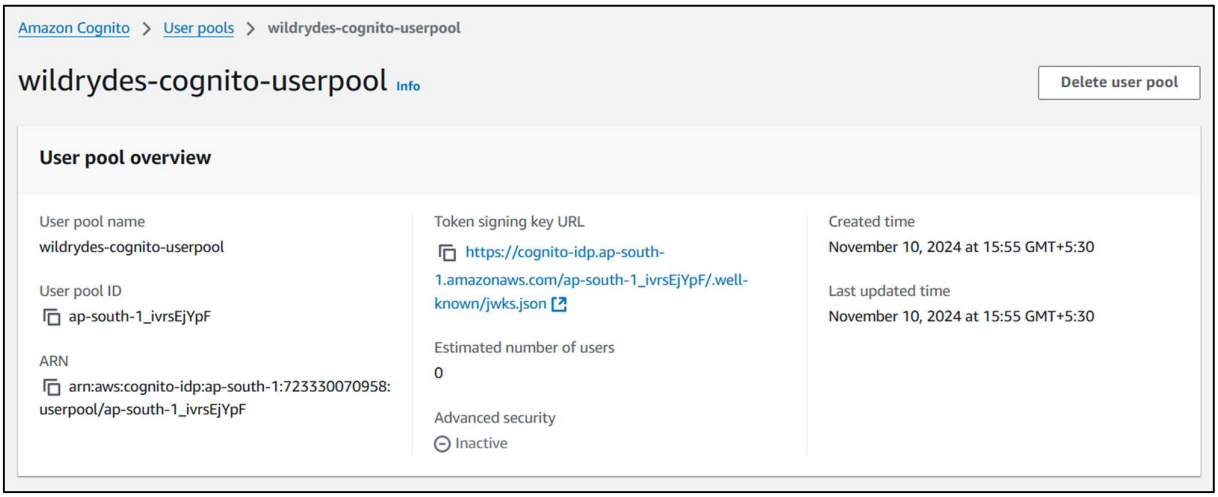
1. Launch Cognito Console
2. Click on “Create User Pool”
3. Select “Cognito user pool sign-in options” as Username

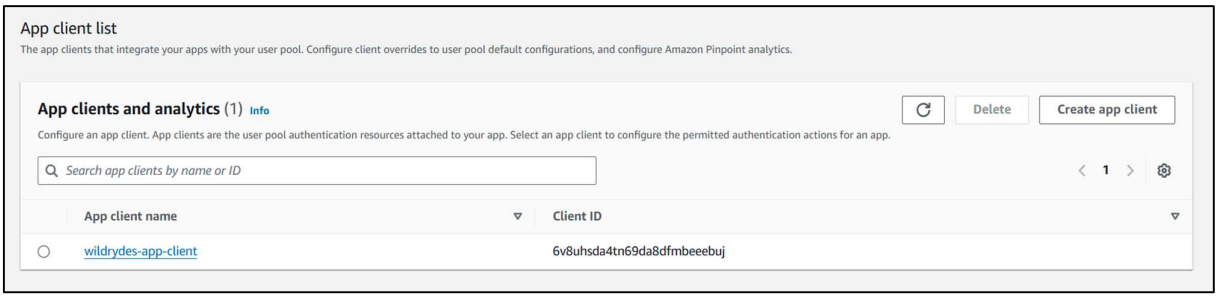


4. Select No MFA in Multi-factor authentication
5. Select Email Provider as “Send email with Cognito”

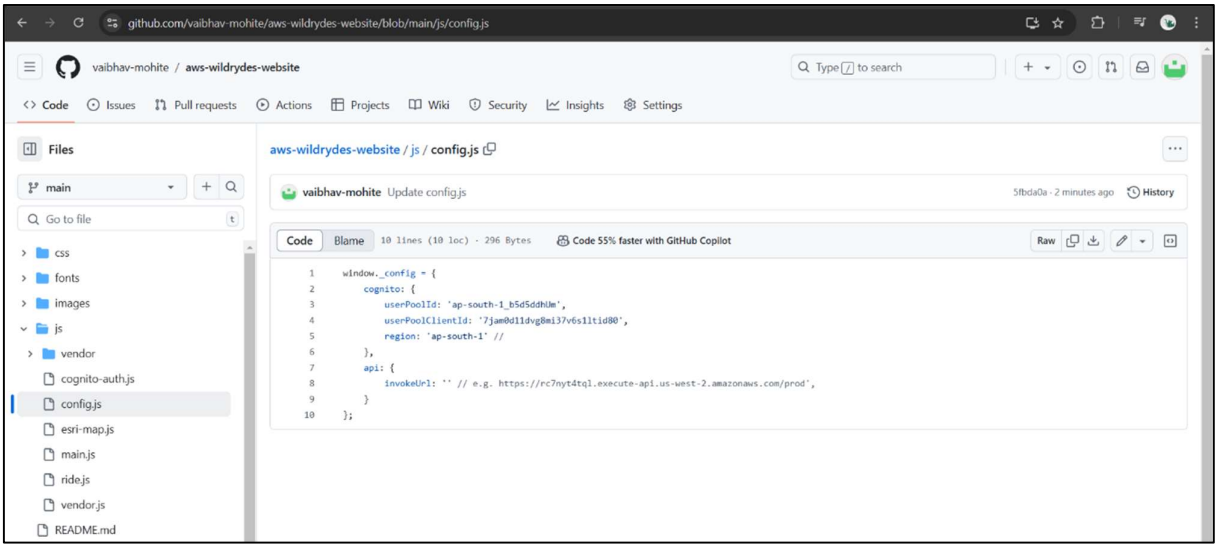


6. Keep rest options as default and Create Userpool
7. Once the Userpool is created, note down User pool ID and App Client ID. App Client ID can be found at User Pool → App integration → App client list

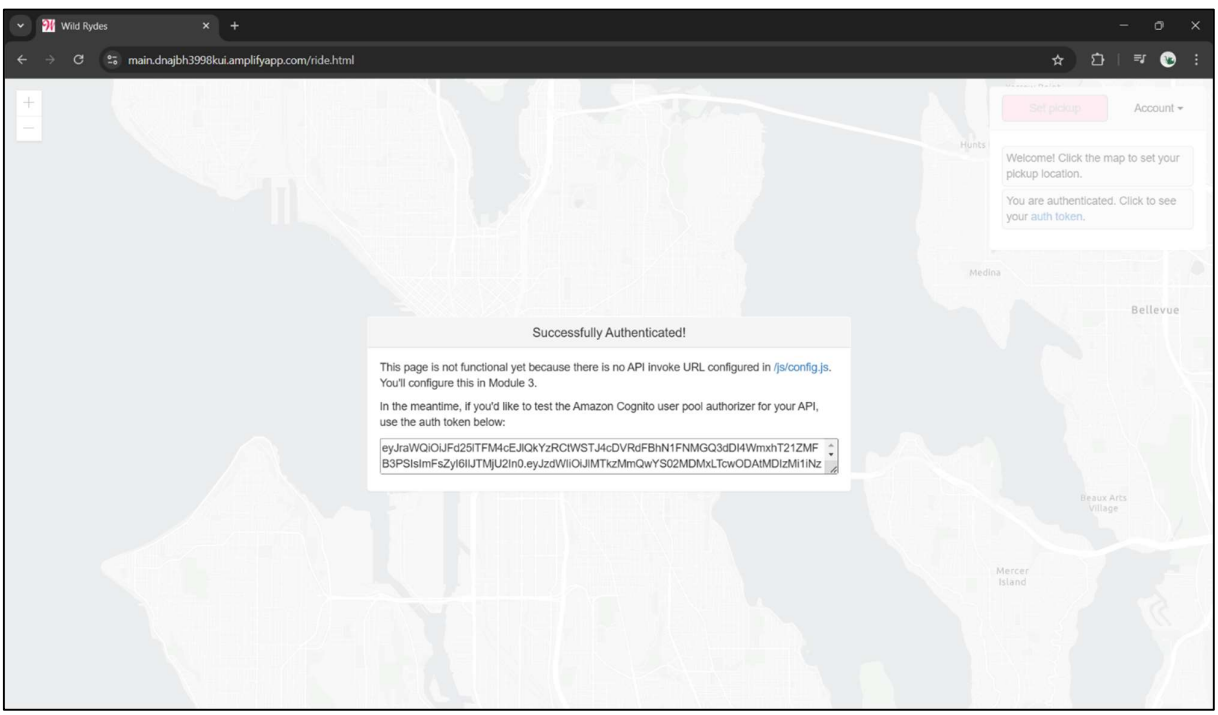




8. Open GitHub Repo. Navigate to js → vendor → config.js
9. Edit UserPoolId, UserPoolClientId and region with the values copied in Step 7 of Cognito.

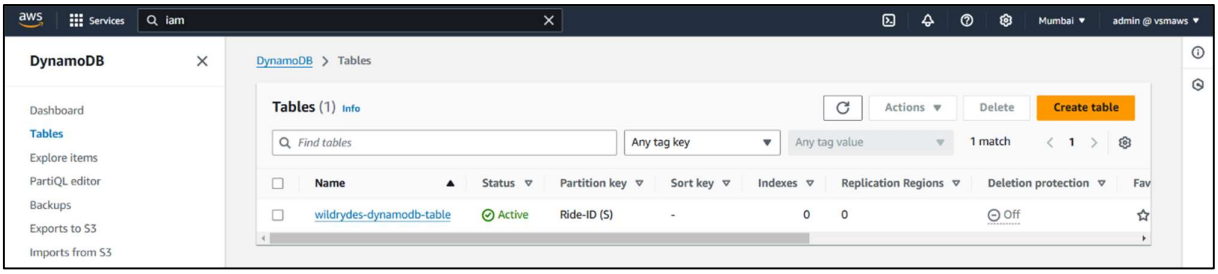


10. Commit the changes. This will start a new Deployment.
11. Once App is Deployed. Open the site. Click on Giddy Up to register a new user. Enter Email ID and Password and verify the same.
12. Login in again. Copy the auth code. It will be used later.



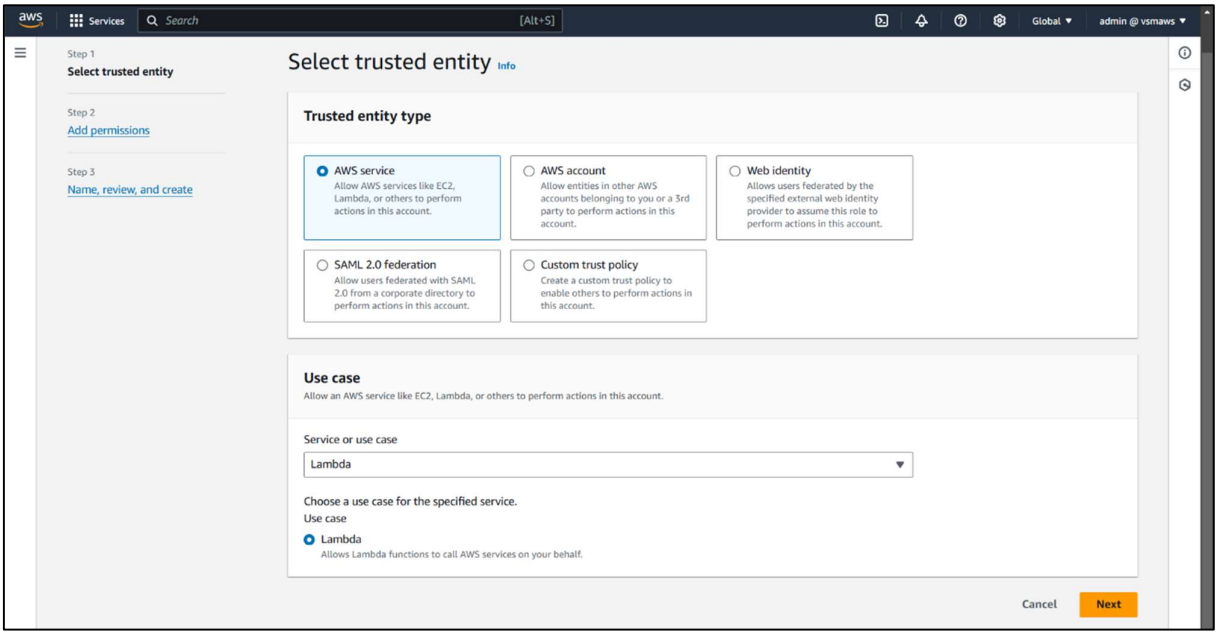
DYNAMODB

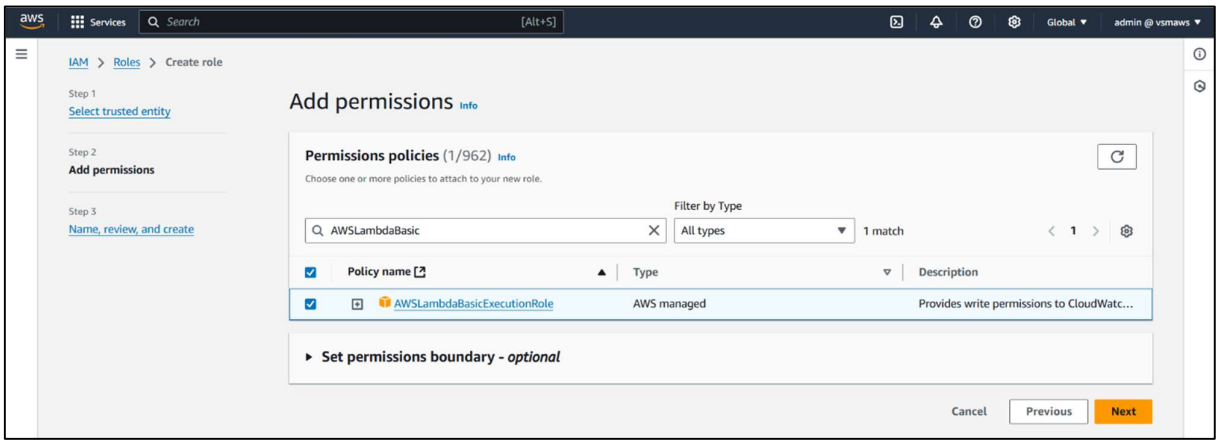
1. Launch DynamoDB Console
2. Create a new Table.
3. Table Name : wildrydes-dynamodb-table
   Partition Key : RideID (String)

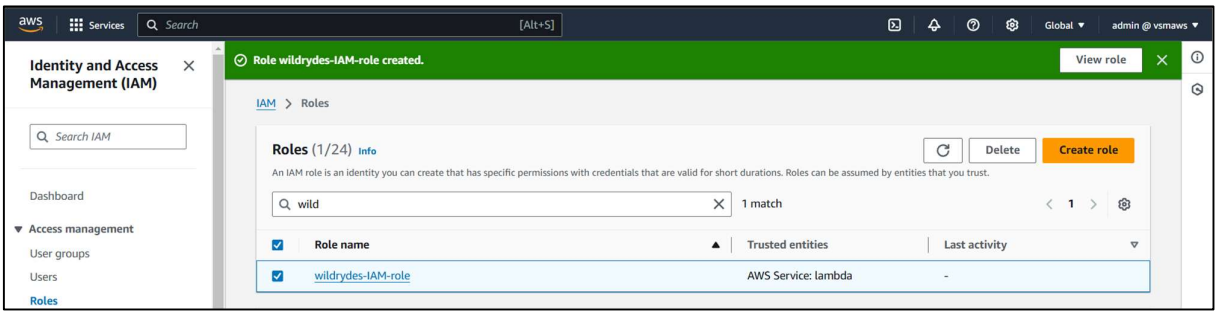


IAM

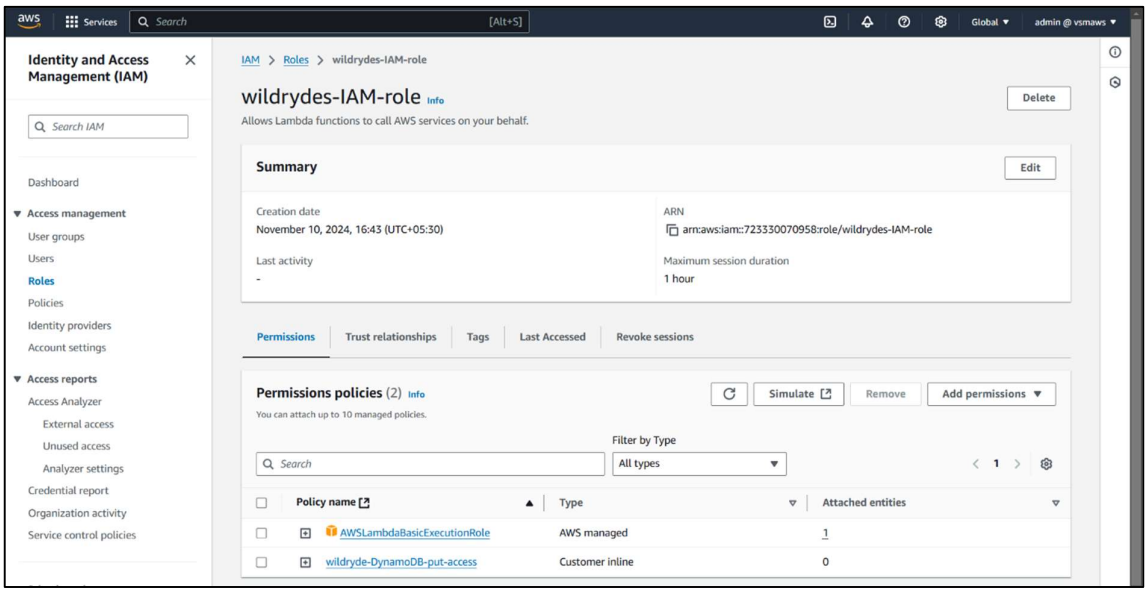
1. Create New Role as seen in Images below.

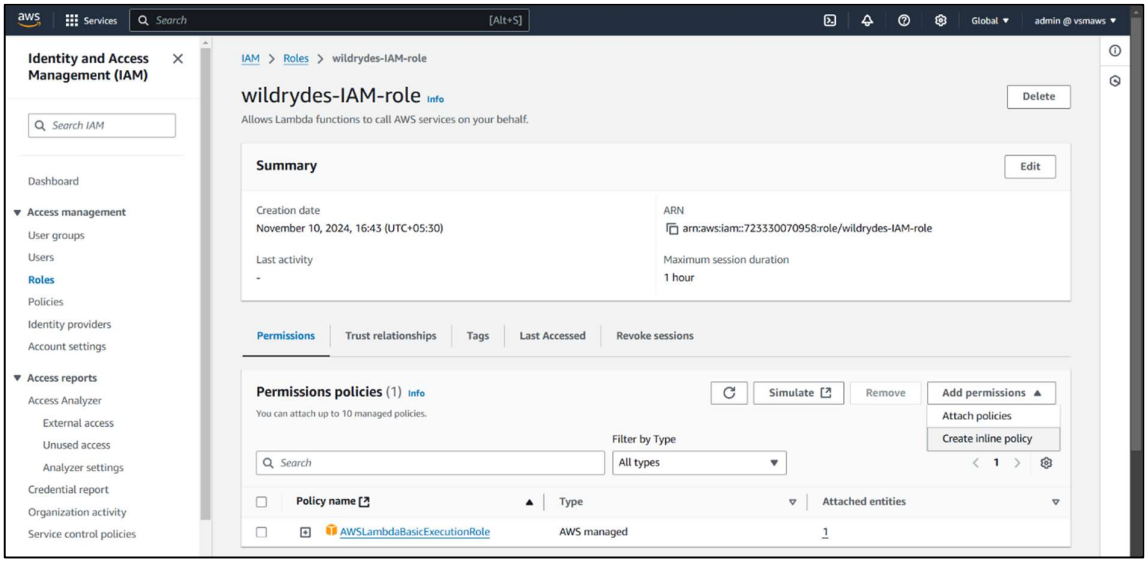


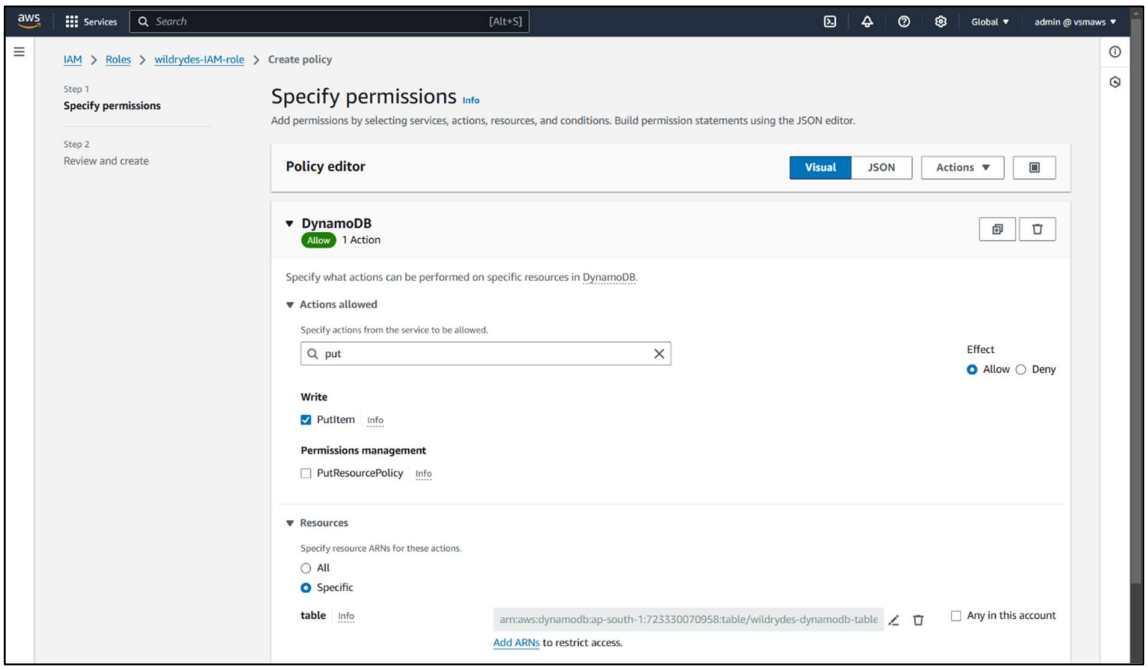




2. Once role is created, add DynamoDB Put Item permission in the IAM Role.

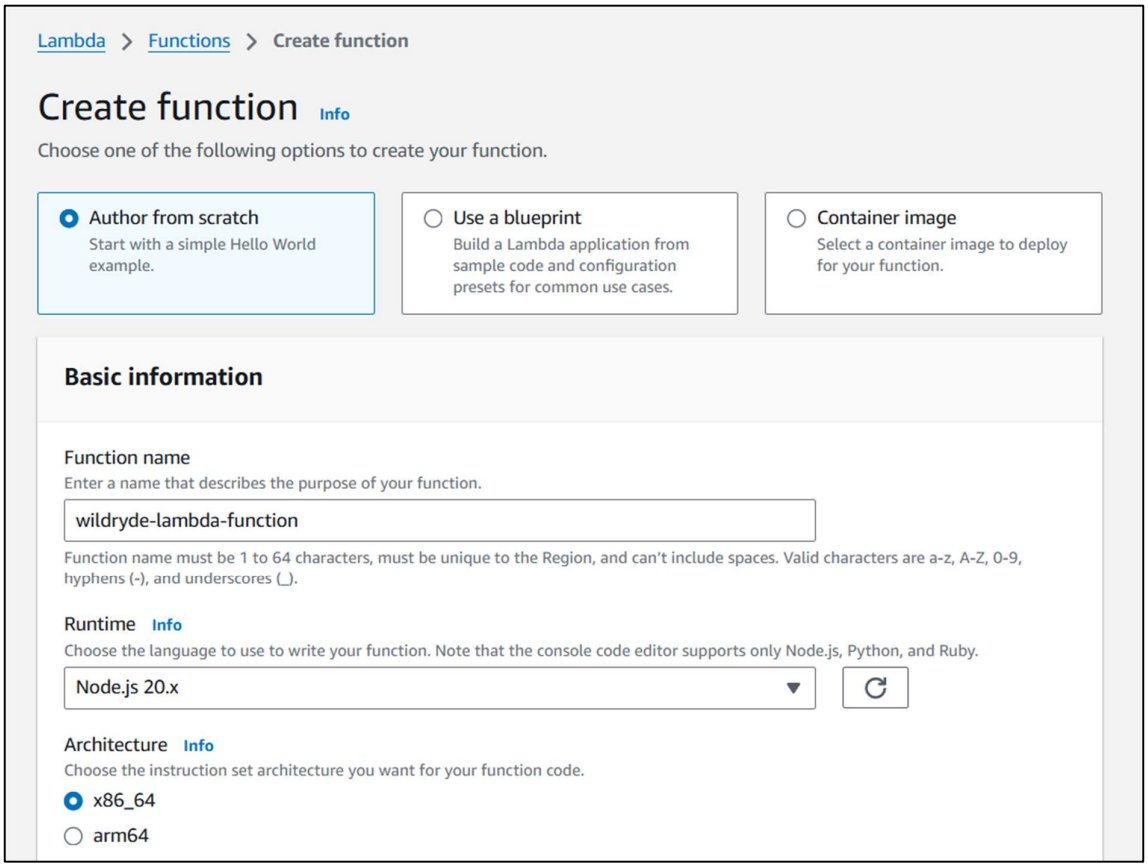




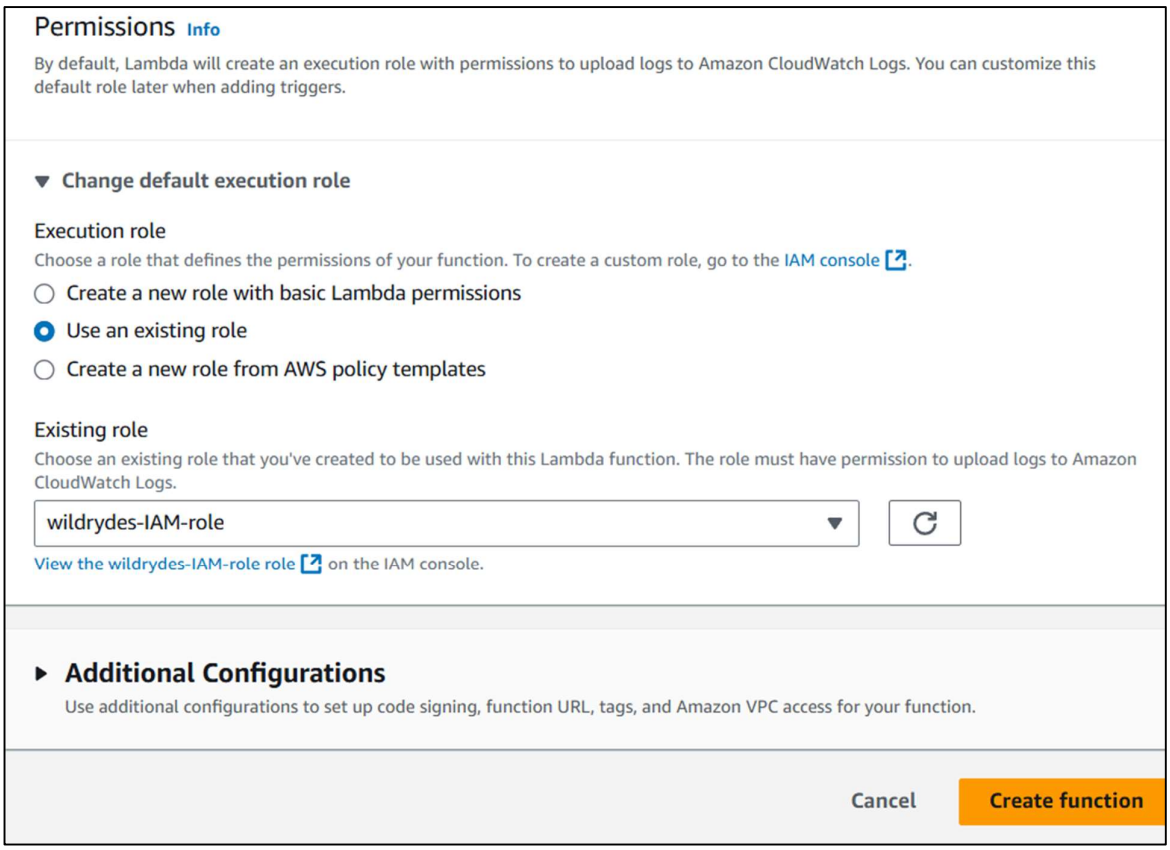


LAMBDA

1. Create a New Lambda Function.



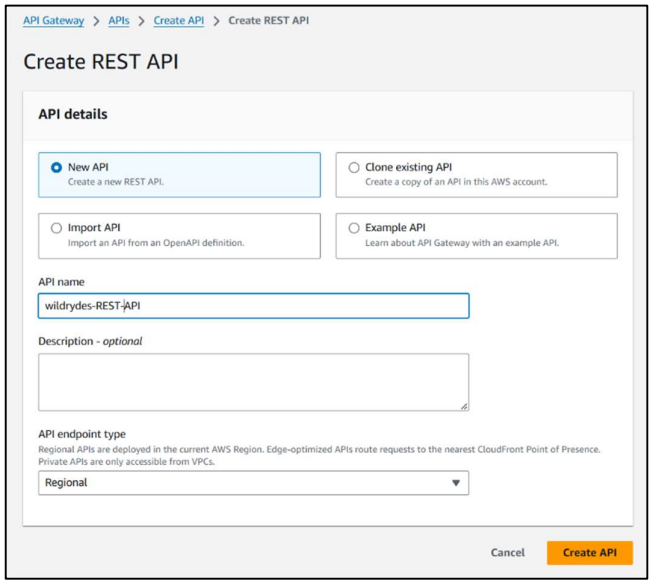
2. Select IAM Role created earlier.



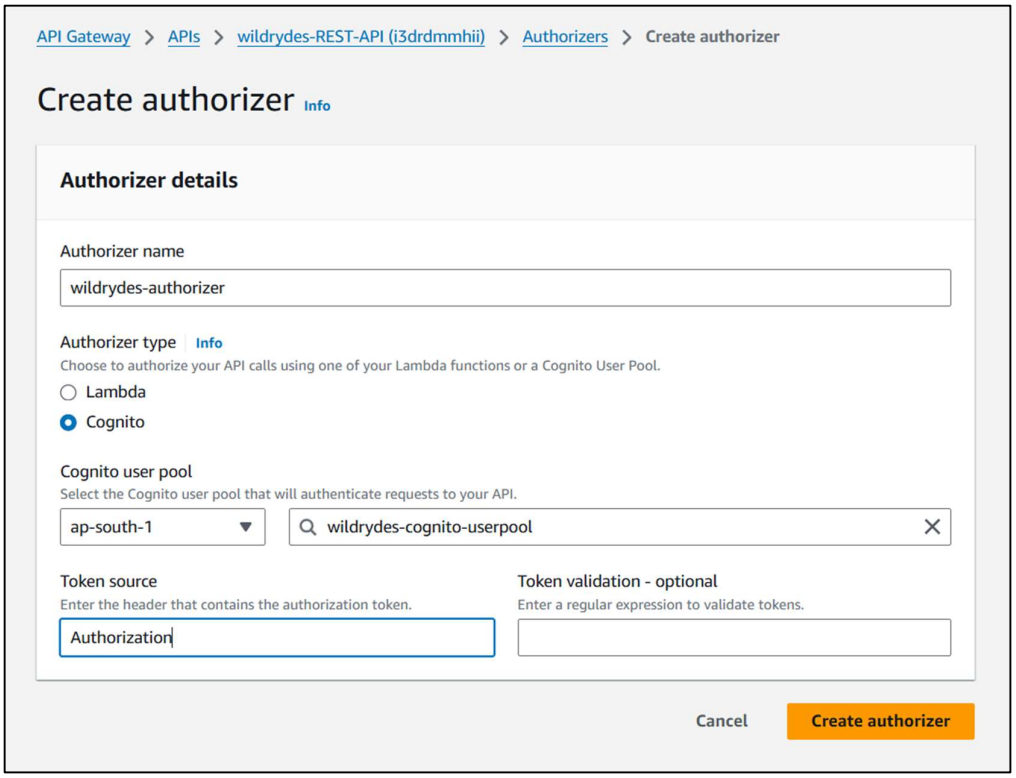
3. Edit the Lambda Code and replace it with Lambda Function Code which could be found out in Readme File of GitHub Repo.

API GATEWAY

1. Click on Create API
2. Select REST API
3. Create a New API named “wildrydes-REST-Api”



4. Click on Authorizers in Left Pane of API Gateway.
5. Click on Create Authorizer
6. Select Authorizer Type as Cognito. Select “wildrydes-cognito-userpool” userPool which was created earlier.
7. Type Token Source as Authorization.



8. To test if authorizer is working as expected, click on newly created Authorizer and paste the token Value which was copied in Step 12 of Cognito.
9. Test the authorizer. The result should output 200 Code.

API Gateway > APIs > wildrydes-REST-API (i3drdmmhii) > Authorizers > wildrydes-authorizer

wildrydes-authorizer

Edit | Delete

Authorizer details

Authorizer ID
gcbr7d

Token source
Authorization

Cognito pool
wildrydes-cognito-userpool - b5d5ddhUm (ap-south-1)

Token validation - optional
None

Test authorizer

Test authorizer

Test your authorizer with a simulated invocation request. Enter an identity token that's provisioned from your Cognito user pool.

Token source
Authorization

Token value
eyJraWQiOiJFd25lTFM4cEJlQkYzRCtWSTJ4cDVRdFE

Authorizer test: wildrydes-authorizer
200
Claims


```
{
  "aud": "7jam0d11dvg8mi37v6s1ltid80",
  "auth_time": "1731236648",
  "cognito:username": "xjaddfif-at-telegmail.com",
  "email": "xjaddfif@telegmail.com",
  "email_verified": "true",
  "event_id": "b53b12d3-38dd-431d-8dcf-07f74198e09e",
  "exp": "Sun Nov 10 12:04:08 UTC 2024",
  "iat": "Sun Nov 10 11:04:08 UTC 2024",
  "iss": "https://cognito-idp.ap-south-1.amazonaws.com/ap-south-1_b5d5ddhUm",
  "jti": "9242e227-9240-4d0f-b03d-d4a77ca4d0c4",
```


10. Click on Resources Tab in Left Pane
11. Click on Create Resource. Enable CORS.

API Gateway > APIs > Resources - wildrydes-REST-API (i3drdmmhii) > 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.

Cancel | Create resource

12. One resource is created, click on “Create Method”Select Method type as POST. Select Lambda Function created earlier.

API Gateway > APIs > Resources - wildrydes-REST-API (i3drdmmhii) > 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.

ap-south-1 | arn:aws:lambda:ap-south-1:723330070958:function:wil

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.

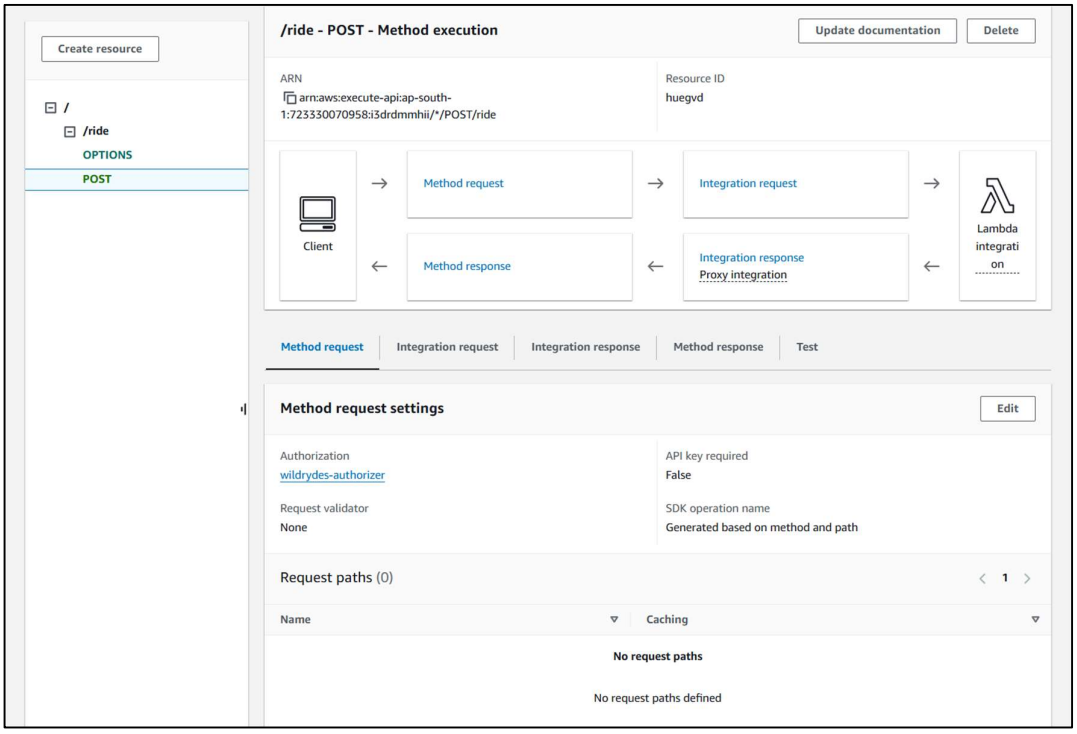
Integration timeout | Info

By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms

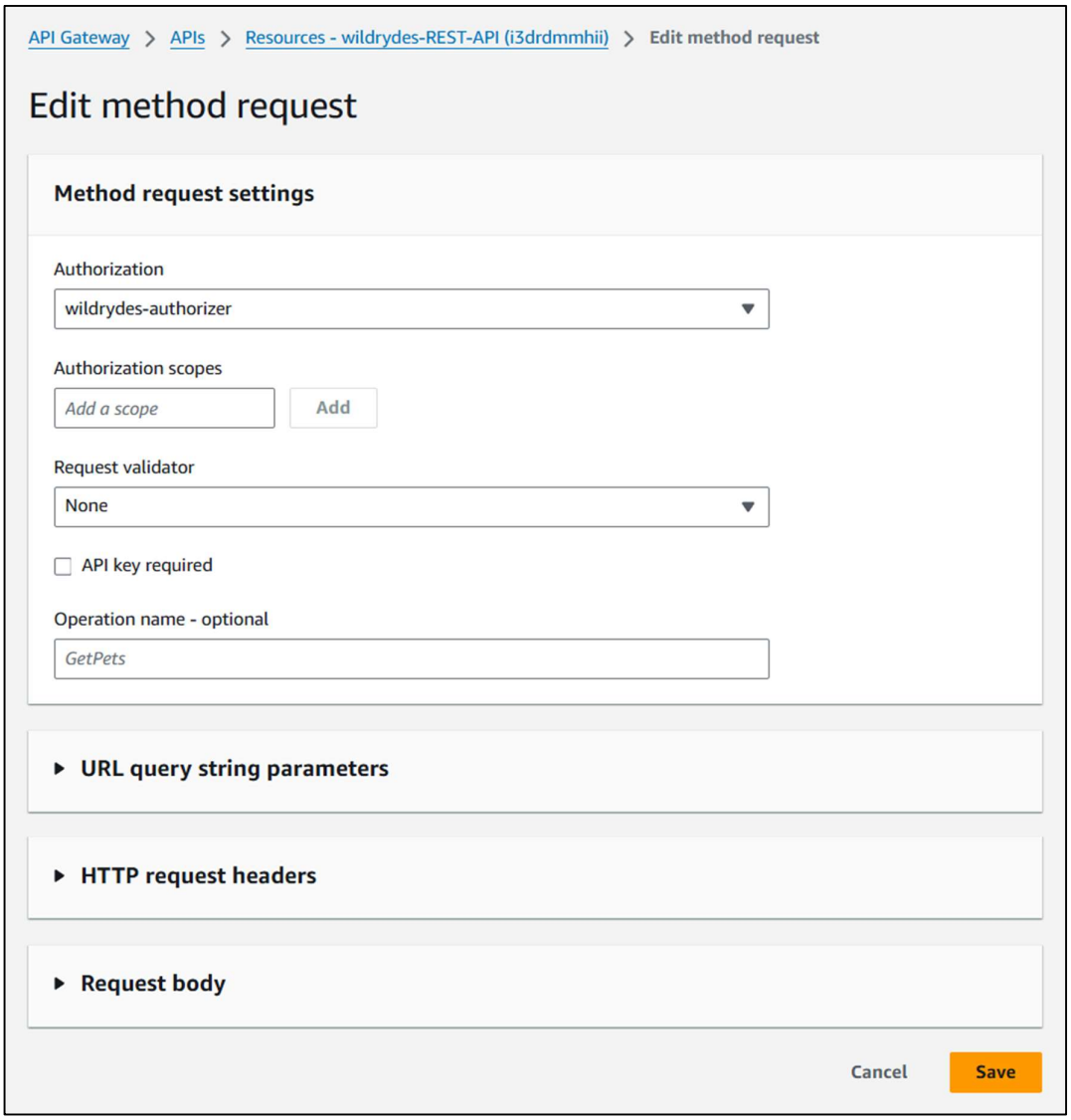
29000

13. Create Method

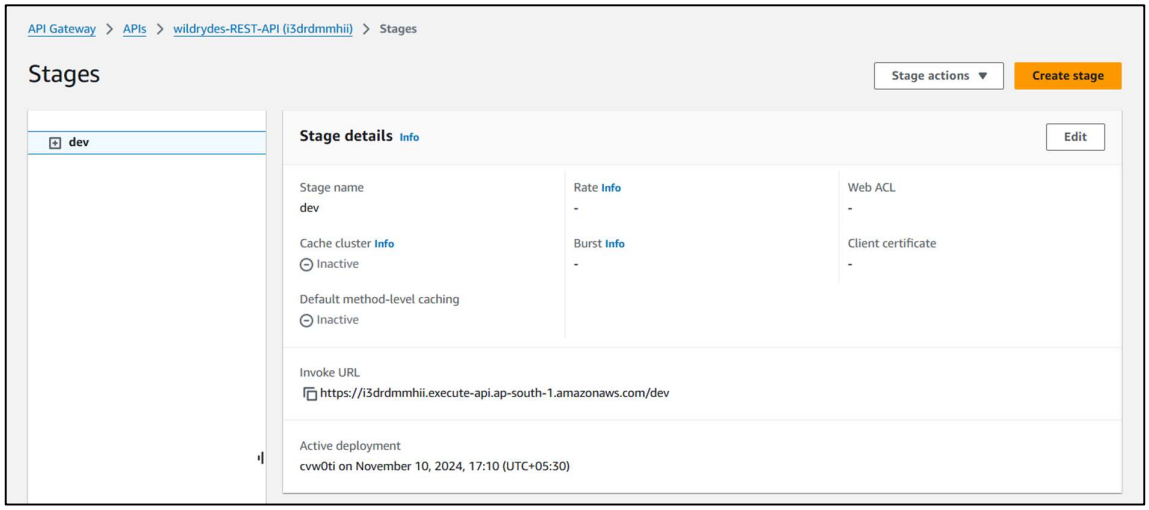
14. Edit Method Request



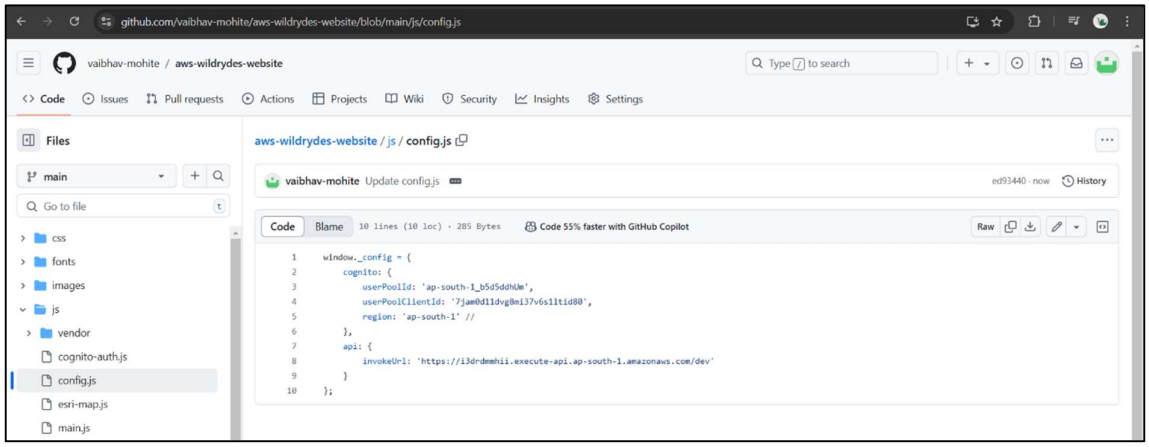
15. Select Authorization as Cognito User Pool Authorizers



16. Save and Deploy the API

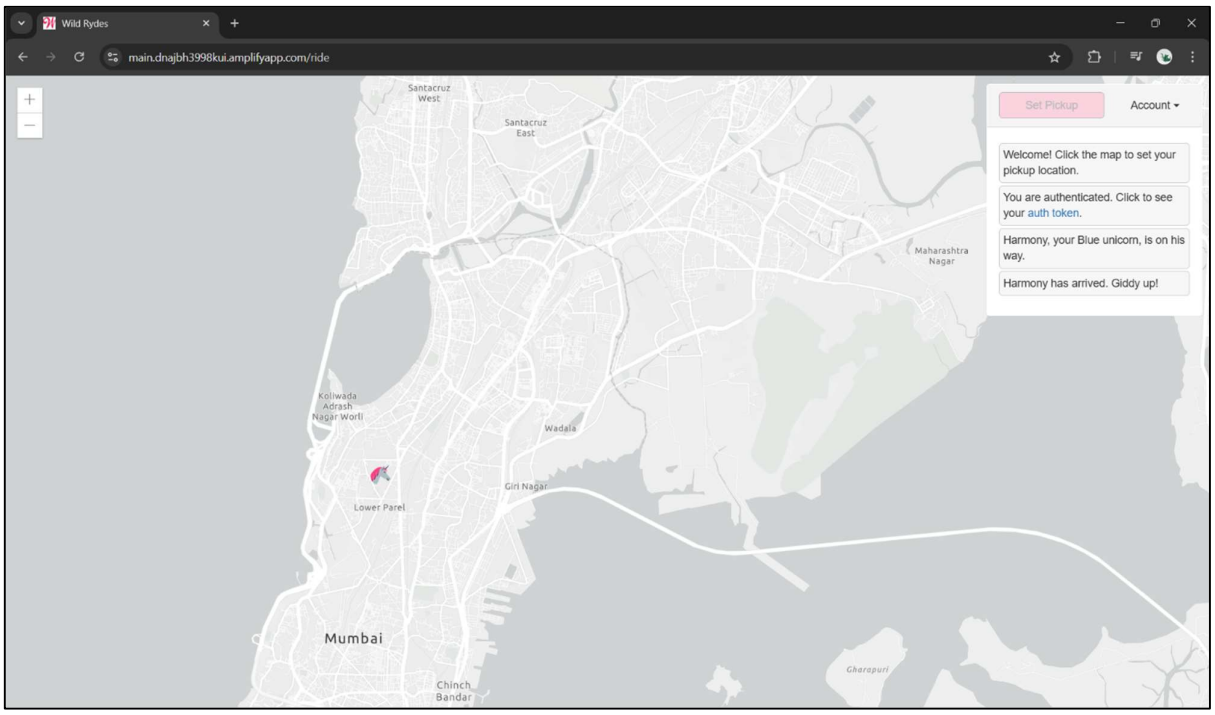


17. Copy the Invoke API and Paste it in GitHub Repo → js → vendor → config.js and paste it in invokeUrl. Commit Changes. These changes will trigger new deployment.



WEBSITE

1. Add /ride.html to the Domain url. This will load the APP. Click on Any Place on the Map and request the unicorn.



Please check "final-website-images" Folder in GitHub Repo for more Images of the Application