

BITS Pilani

# Security Fundamentals for Cloud

Course code: (S2-23\_CCZG504)

Assignment – 1-2

Submitted by:

Sruthi Krishnamurthy(2023mt03003)



May 2024

## 1. Creating Cloud 9 environment named serverless

The screenshot shows the AWS Cloud9 control panel. A green success message at the top right states: "Successfully created serverless. To get the most out of your environment, see [Best practices for using AWS Cloud9](#)". Below this, the "Environments" section displays a table with one row:

Name	Cloud9 IDE	Environment type	Connection	Permissions	Owner ARN
serverless	<a href="#">Open</a>	EC2 instance	AWS Systems Manager (SSM)	Owner	arn:aws:sts::769935406560:federated-user/2023mt03003@wil1714582894815

## 2. creating repository

The screenshot shows the AWS Cloud9 IDE interface. In the terminal window, the user runs the command to create a new repository:

```
aws - "ip-172-31-84-115.e x" Immediate x +  
2023mt03003@wil1714582894815:~/environment $ aws codecommit create-repository \  
--repository-name wild-rydes  
{  
    "repositoryMetadata": {  
        "accountId": "769935406560",  
        "repositoryId": "f0650c6f-cdd8-4715-859c-223325384f2a",  
        "repositoryName": "wild-rydes",  
        "lastModifiedDate": "2024-05-01T17:09:42.260000+00:00",  
        "creationDate": "2024-05-01T17:09:42.260000+00:00",  
        "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/repos/wild-rydes",  
        "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/wild-rydes",  
        "arn": "arn:aws:codecommit:us-east-1:769935406560:wild-rydes",  
        "kmsKeyId": "arn:aws:kms:us-east-1:769935406560:key/6bdb2382-97f9-4db5-b8c2-a3829fefe687"  
    }  
}  
2023mt03003@wil1714582894815:~/environment $
```

### 3. Repo created

The screenshot shows the AWS CodeCommit service in the AWS Management Console. On the left, a navigation pane titled 'CodeCommit' lists various services like Source, Artifacts, Build, Deploy, Pipeline, and Settings. The main area is titled 'Repositories' and shows a single repository named 'wild-rydes'. The repository details are as follows:

Name	Description	Last modified	Clone URL	AWS KMS Key
wild-rydes	-	2 minutes ago	HTTPS SSH HTTPS (GRC)	arn:aws:kms:us-east-1:769935406560:key/6b4b2382-97f9-4db5-b8c2-a3829fefe687

### 4. Cloning github branch

The screenshot shows the AWS Cloud9 IDE interface. On the left, a file explorer window shows a directory structure with 'serverless - /home' and 'aws-serverless-webapp' containing a 'README.md' file. The main workspace is a terminal window titled 'bash - "ip-172-31-84-115.x"' showing the command to clone a GitHub repository:

```
git clone https://github.com/aws-samples/aws-serverless-webapp-workshop.git
```

The terminal output shows the progress of cloning the repository from GitHub into the local environment.

### 5. Folder created with necessary files

The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file explorer sidebar with a tree view of a project named 'serverless - hon'. The main area is a terminal window titled 'bash - \*ip-172-31-84-115.x' showing the command:

```
git clone https://github.com/aws-samples/aws-serverless-webapp-workshop.git
```

The output of the command is displayed below:

```
Cloning into 'aws-serverless-webapp-workshop'...
remote: Enumerating objects: 994, done.
remote: Counting objects: 100% (223/223), done.
remote: Compressing objects: 100% (121/121), done.
remote: Total 994 (delta 73), reused 186 (delta 64), pack-reused 771
Receiving objects: 100% (994/994), 58.68 MiB | 24.78 MiB/s, done.
Resolving deltas: 100% (193/193), done.
Updating files: 100% (515/515), done.
```

## 6. Creating a branch

The screenshot shows the AWS Cloud9 IDE interface. The terminal window shows the same cloning command as before:

```
git clone https://github.com/aws-samples/aws-serverless-webapp-workshop.git
```

Output:

```
Cloning into 'aws-serverless-webapp-workshop'...
remote: Enumerating objects: 994, done.
remote: Counting objects: 100% (223/223), done.
remote: Compressing objects: 100% (121/121), done.
remote: Total 994 (delta 73), reused 186 (delta 64), pack-reused 771
Receiving objects: 100% (994/994), 58.68 MiB | 24.78 MiB/s, done.
Resolving deltas: 100% (193/193), done.
Updating files: 100% (515/515), done.
```

Then, the user runs:

```
cd aws-serverless-webapp-workshop
sudo yum install git-subtree -y
```

Output:

```
git subtree split -P resources/code/WildrydesVue -b WildrydesVue
Last metadata expiration check: 0:08:47 ago on Wed May 1 17:05:44 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
git-subtree	noarch	2.40.1-1.amzn2023.0.1	amazonlinux	39 k

## 7. Creating a folder

```

Running scriptlet: git-subtree-2.40.1-1.amzn2023.0.1.noarch
Verifying      : git-subtree-2.40.1-1.amzn2023.0.1.noarch
-----
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:
Version 2023.4.20240429:
  Run the following command to upgrade to 2023.4.20240429:

  dnf upgrade --releasever=2023.4.20240429

Release notes:
  https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.4.20240429.html

-----
Installed:
git-subtree-2.40.1-1.amzn2023.0.1.noarch

Complete!
2023mt03003@will1714582894815:~/environment/aws-serverless-webapp-workshop (main) $ git subtree split -P resources/code/WildRydesVue -b WildRydesVue
Created branch 'WildRydesVue'
ad15e140c5e52f07a9e4012645e813901943cb
2023mt03003@will1714582894815:~/environment/aws-serverless-webapp-workshop (main) $ mkdir ..wild-rydes && cd ..wild-rydes
2023mt03003@will1714582894815:~/environment/wild-rydes $ 

```

## 8. Git pull. We can see the folder structure

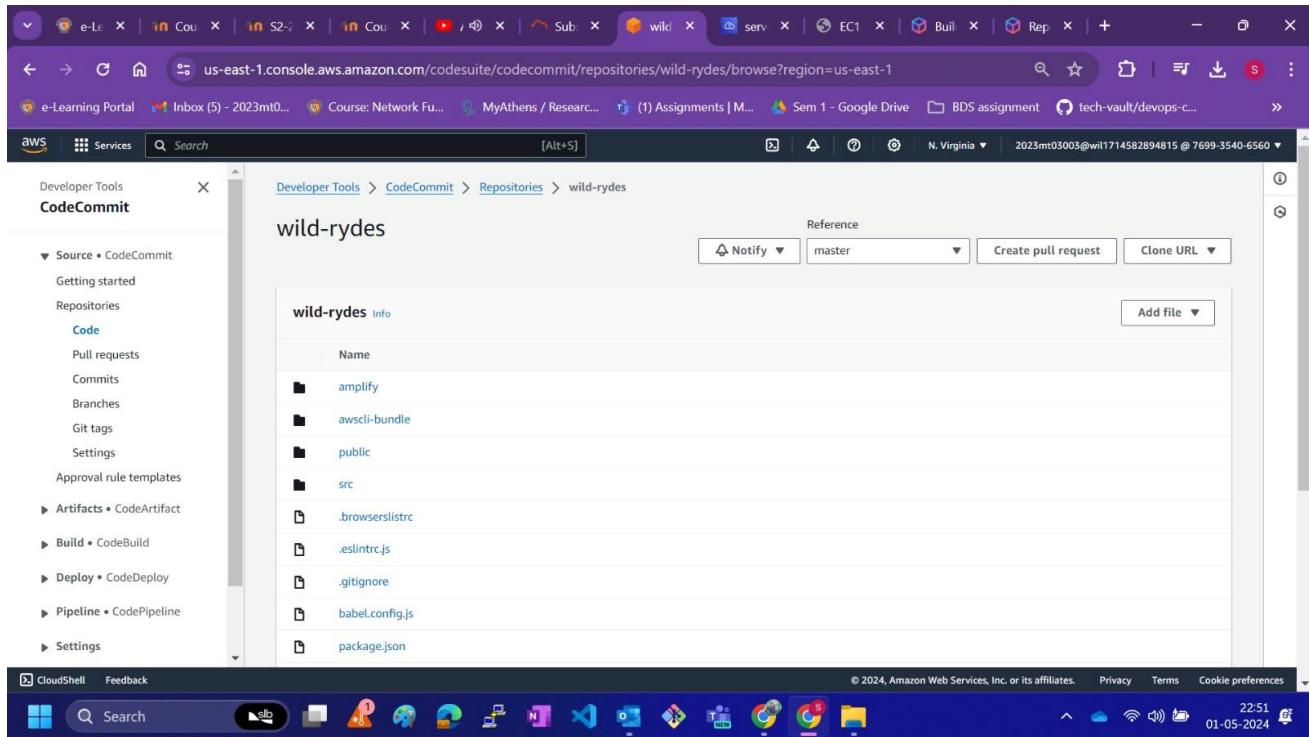
```

Go to Anything (Ctrl-P)
serverless - /home/ec2-user/environment/wild-rydes
aws
aws-serverless-webapp
wild-rydes
  amplify
  awscii-bundle
  public
src
  babel.config.js
  package.json
  postcss.config.js
  README.md
  vue.config.js
  README.md

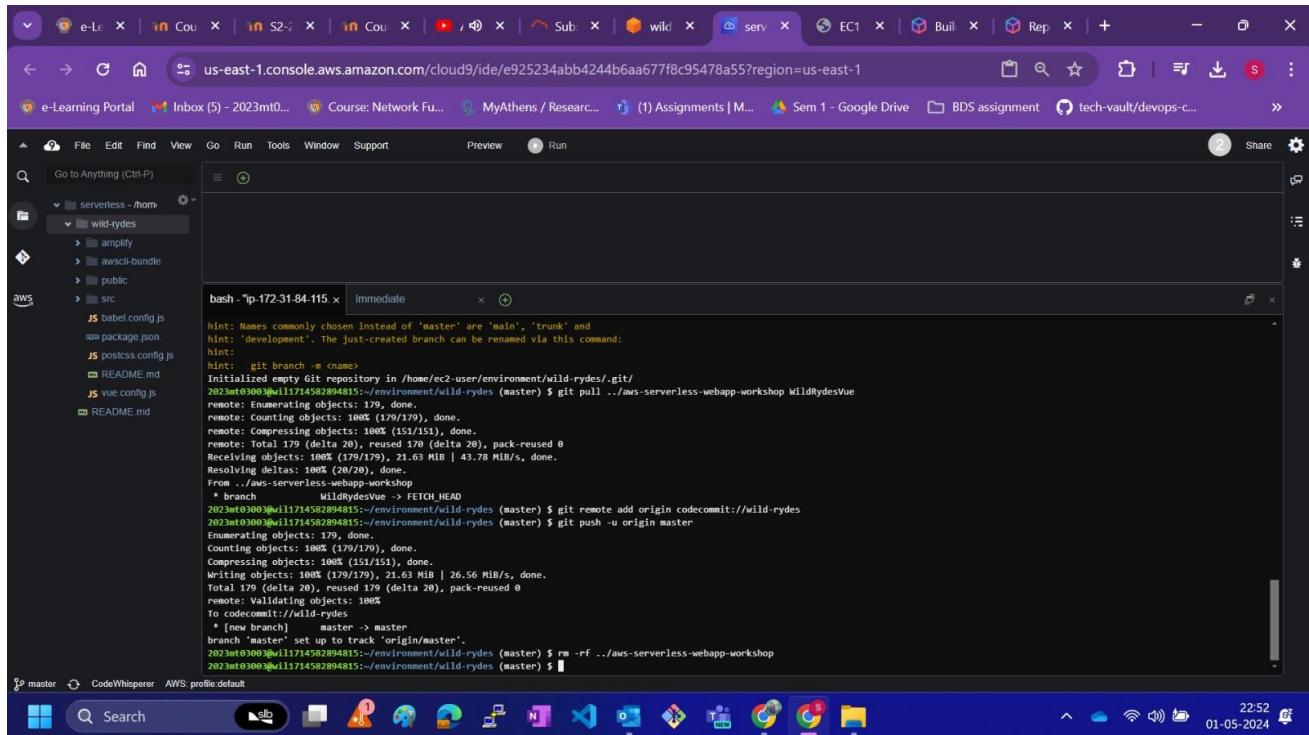
Complete!
2023mt03003@will1714582894815:~/environment/aws-serverless-webapp-workshop (main) $ git subtree split -P resources/code/WildRydesVue -b WildRydesVue
Created branch 'WildRydesVue'
ad15e140c5e52f07a9e4012645e813901943cb
2023mt03003@will1714582894815:~/environment/aws-serverless-webapp-workshop (main) $ mkdir ..wild-rydes && cd ..wild-rydes
2023mt03003@will1714582894815:~/environment/wild-rydes $ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/environment/wild-rydes/.git/
2023mt03003@will1714582894815:~/environment/wild-rydes (master) $ git pull ../aws-serverless-webapp-workshop WildRydesVue
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (151/151), done.
remote: Total 179 (delta 20), reused 170 (delta 20), pack-reused 0
Receiving objects: 100% (179/179), 21.63 MiB | 43.78 MiB/s, done.
Resolving deltas: 100% (20/20), done.
From ../aws-serverless-webapp-workshop
 * branch      WildRydesVue -> FETCH_HEAD
2023mt03003@will1714582894815:~/environment/wild-rydes (master) $ 

```

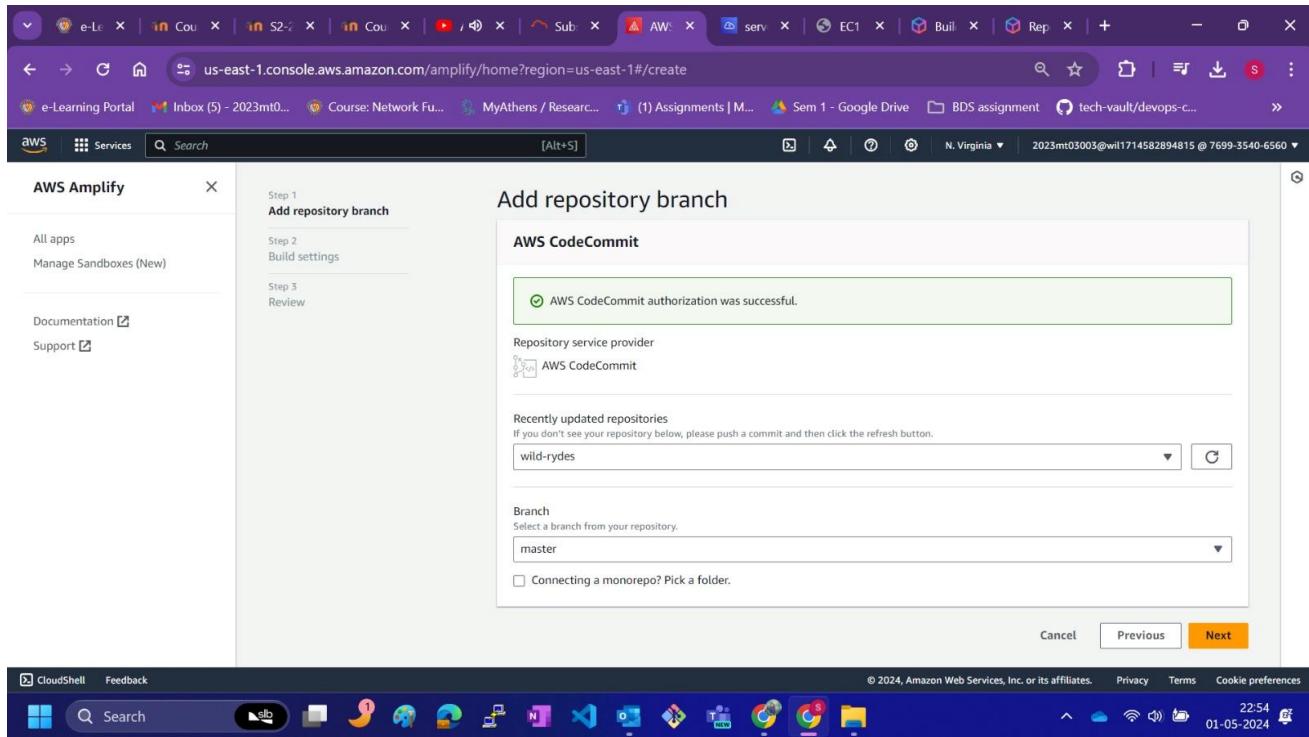
## 9. All files have come up on the code commit repo



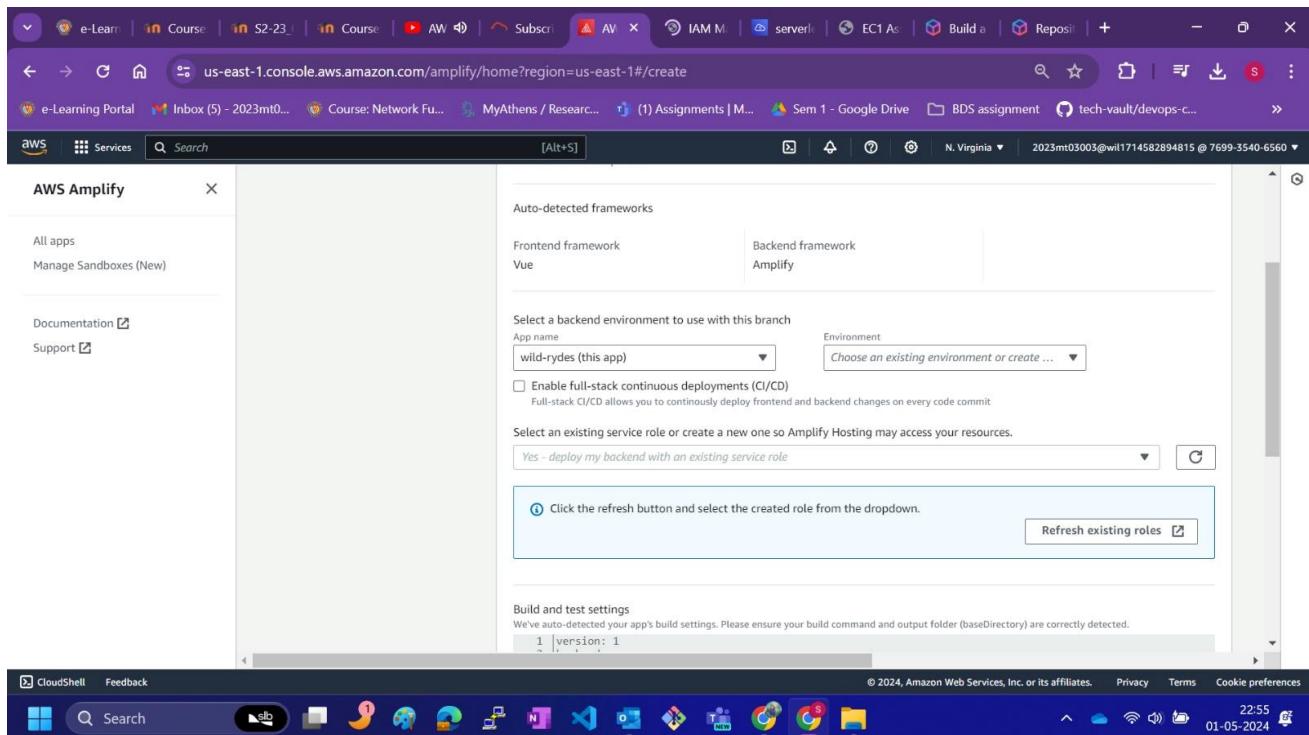
10. Folder removed. Source code now is cloud commit and linked with cloud 9



11. Go to AWS amplify and link to the repo



## 12. Creating a new role



## 13. Adding read only permissions to amplify

Add permissions

Permissions policies (1)

Policy name: AdministratorAccess-Amplify

Type: AWS managed

Set permissions boundary - optional

#### 14. Amplify role created

Identity and Access Management (IAM)

Role amplifyconsole-backend-role created.

Role name	Trusted entities	Last activity
amplifyconsole-backend-role	AWS Service: amplify	-
AWSCloud9SSMAccessRole	AWS Service: ec2, and 1 more.	21 minutes ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked)	21 minutes ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
nl-admin-dont-del	Account: 532344024030	2 days ago

Roles Anywhere

Manage

#### 15. Now we can see the role

The screenshot shows the AWS Amplify console interface. On the left, there's a sidebar with options like 'All apps' and 'Documentation'. The main area is titled 'AWS Amplify' and shows a step-by-step process for creating a new app. Step 1: 'Add repository branch' (selected), Step 2: 'Build settings', Step 3: 'Review'. In the 'Review' section, under 'Repository details', it shows a repository service 'AWS CodeCommit', repository 'wild-rydes', and branch 'master'. Under 'Branch environment', it shows 'prod' and 'Application root'. In the 'App settings' section, it shows 'App name: wild-rydes', 'Framework: Vue - Amplify', 'Build image: Using default image', and 'Build settings: Auto-detected settings will be used'. A note says 'Environment variables: None'. At the bottom, there are 'Cancel', 'Previous', and 'Save and deploy' buttons.

## 16. Saving and deploying

The screenshot shows the AWS Amplify console interface during the deployment process. The status bar at the top says 'Creating app: wild-rydes in progress...'. The main area is titled 'AWS Amplify' and shows a step-by-step process for creating a new app. Step 1: 'Add repository branch' (selected), Step 2: 'Build settings', Step 3: 'Review'. In the 'Review' section, under 'Repository details', it shows a repository service 'AWS CodeCommit', repository 'wild-rydes', and branch 'master'. Under 'Branch environment', it shows 'prod' and 'Application root'. In the 'App settings' section, it shows 'App name: wild-rydes', 'Framework: Vue - Amplify', 'Build image: Using default image', and 'Build settings: Auto-detected settings will be used'. A note says 'Environment variables: None'. At the bottom, there are 'Cancel', 'Previous', and 'Save and deploy' buttons.

## 17. Deploying- provisioning

## 18. Building

```

229 Receiving objects: 83% (149/179)
230 Receiving objects: 84% (151/179)
231 Receiving objects: 85% (153/179)
232 Receiving objects: 86% (154/179)
233 Receiving objects: 87% (156/179)

```

## 19. Modifying title.html in the source code and git push new modification

```

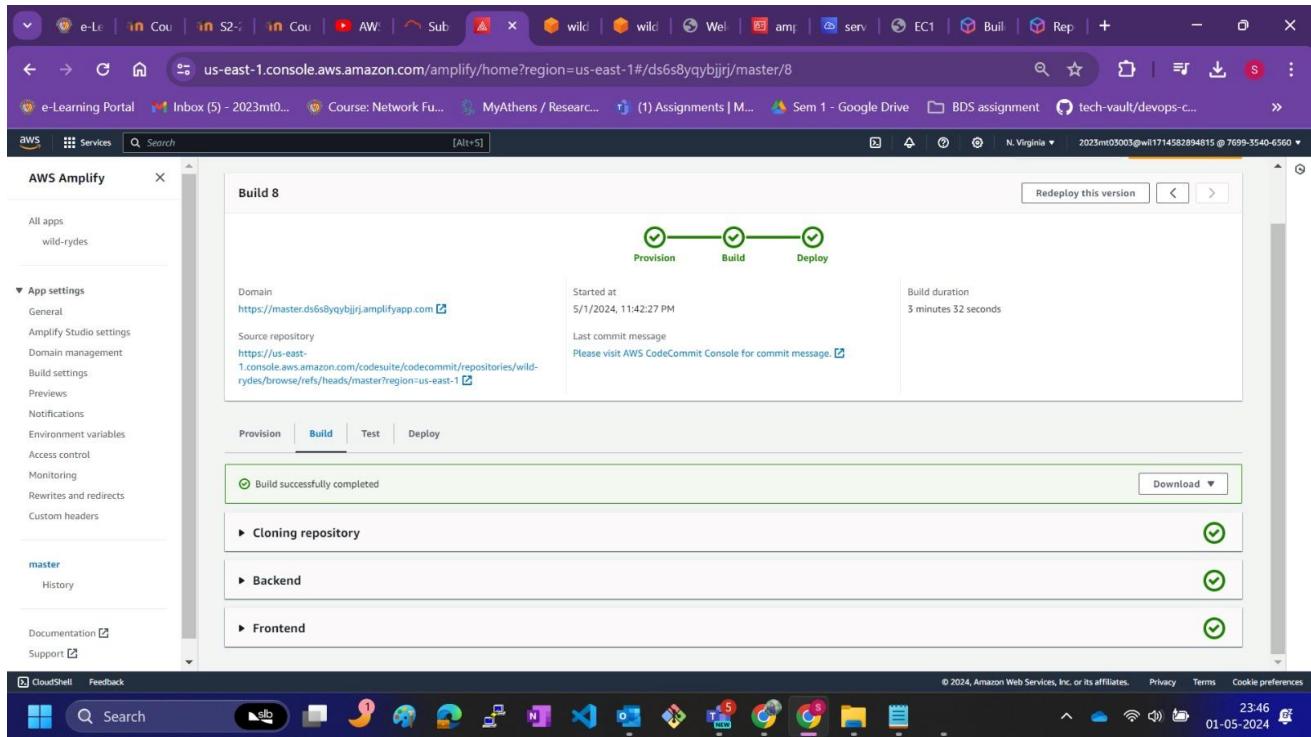
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="https://aws.amazon.com/favicon.ico">
    <title>SFC Assignment-Srujan</title>
  </head>
  <body>
    <link rel="stylesheet" href="/css/front.css">
    <link rel="stylesheet" href="/css/main.css">
    <div class="page-home">
      <strong>We're sorry but wildrydes doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
      <script>
        // ...
      </script>
      <div id="app"></div>
      <div>built files will be auto injected -->
    </div>
  </body>
</html>

```

## 20. Git commits in code commit

Commit ID	Commit message	Commit date	Authored date	Author	Committer	Actions
3e5c58ed	modified package.json	3 minutes ago	3 minutes ago	EC2 Default User	EC2 Default User	<a href="#">Copy ID</a> <a href="#">Browse</a>
1d95dc54	update title	8 minutes ago	8 minutes ago	EC2 Default User	EC2 Default User	<a href="#">Copy ID</a> <a href="#">Browse</a>
a9d15e14	fixed footer. removed 'html'	3 years ago	3 years ago	George Seib	George Seib	<a href="#">Copy ID</a> <a href="#">Browse</a>
63751242	Reverted version changes.	3 years ago	3 years ago	James Beswick	GitHub	<a href="#">Copy ID</a> <a href="#">Browse</a>
ada48501	Updated lint rules	3 years ago	3 years ago	Beswick	Beswick	<a href="#">Copy ID</a> <a href="#">Browse</a>
6ada22b1	Updated versions	3 years ago	3 years ago	Beswick	Beswick	<a href="#">Copy ID</a> <a href="#">Browse</a>
9a07d1c0	bumping Axios version	3 years ago	3 years ago	Chris Munns	GitHub	<a href="#">Copy ID</a> <a href="#">Browse</a>
3212a811	Corrected .gitignore in root that was blocking required Vue public/ directory	4 years ago	4 years ago	Rob Sutter	Rob Sutter	<a href="#">Copy ID</a> <a href="#">Browse</a>

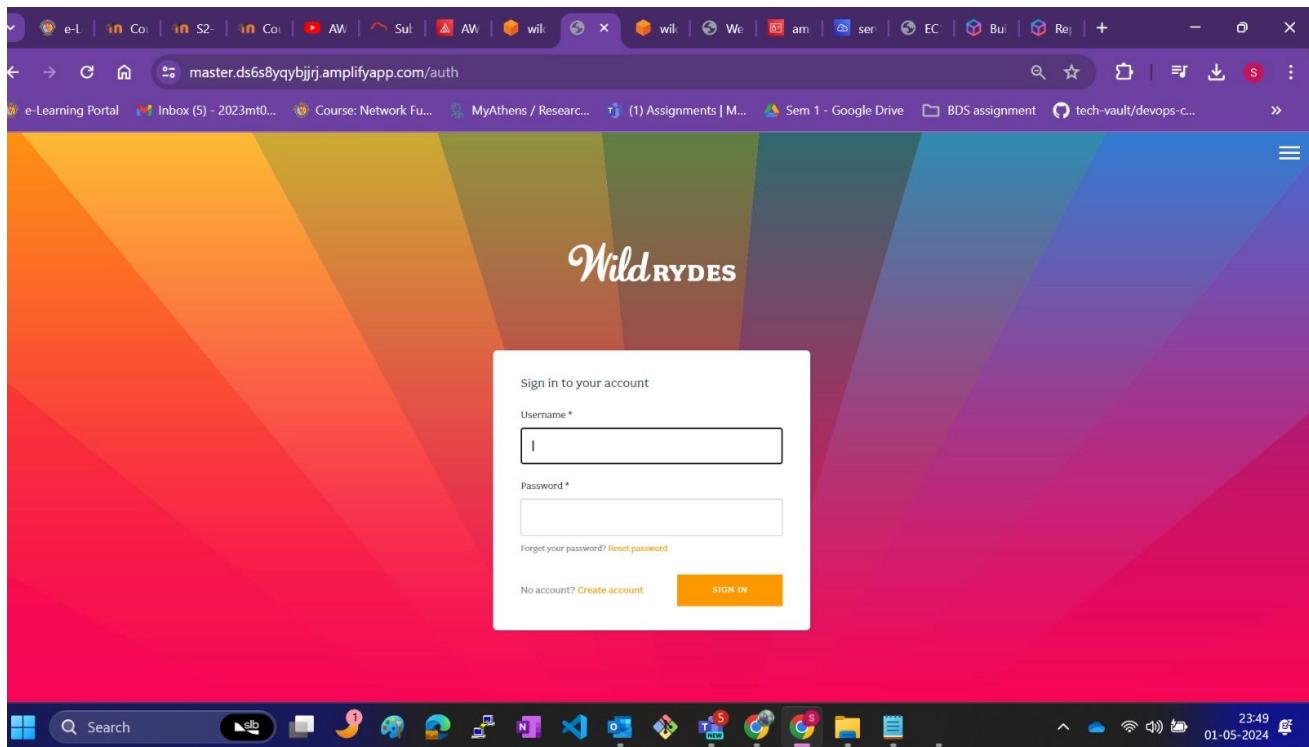
## 21. Build successful



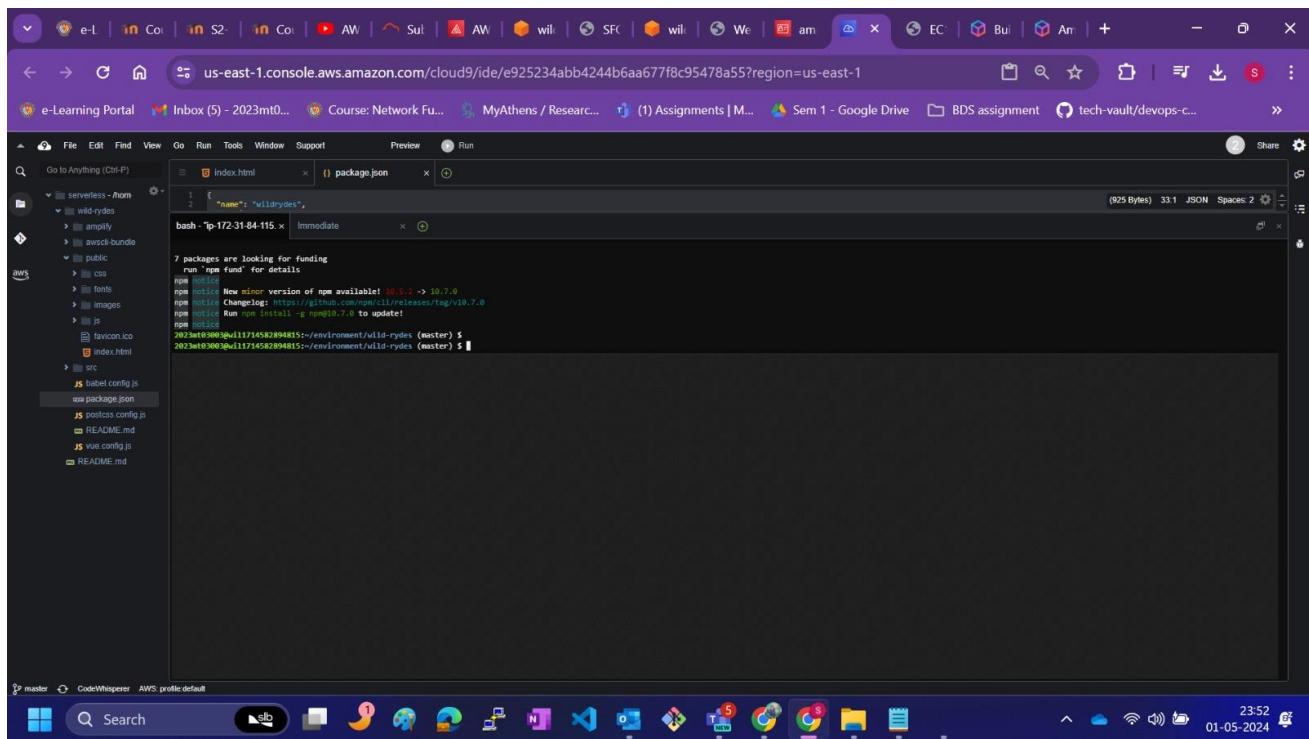
## 22. Website



## 23. Sign in page



## 24. Installing amplify



## 25. Adding amplify authentication

```
node -lp-172-31-84-115: ~environment/wild-rydes (master) $ amplify add auth
Using service: Cognito, provided by: awsCloudFormation

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? (Use arrow keys)
    Default configuration with Social Provider (Federation)
    Manual configuration
    I want to learn more. [
```

## 26. Configure cognito and git commit

```
node -lp-172-31-84-115: ~environment/wild-rydes (master) $ amplify add auth
Using service: Cognito, provided by: awsCloudFormation

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? (Use arrow keys)
    Default configuration with Social Provider (Federation)
    Manual configuration
    I want to learn more. [
```

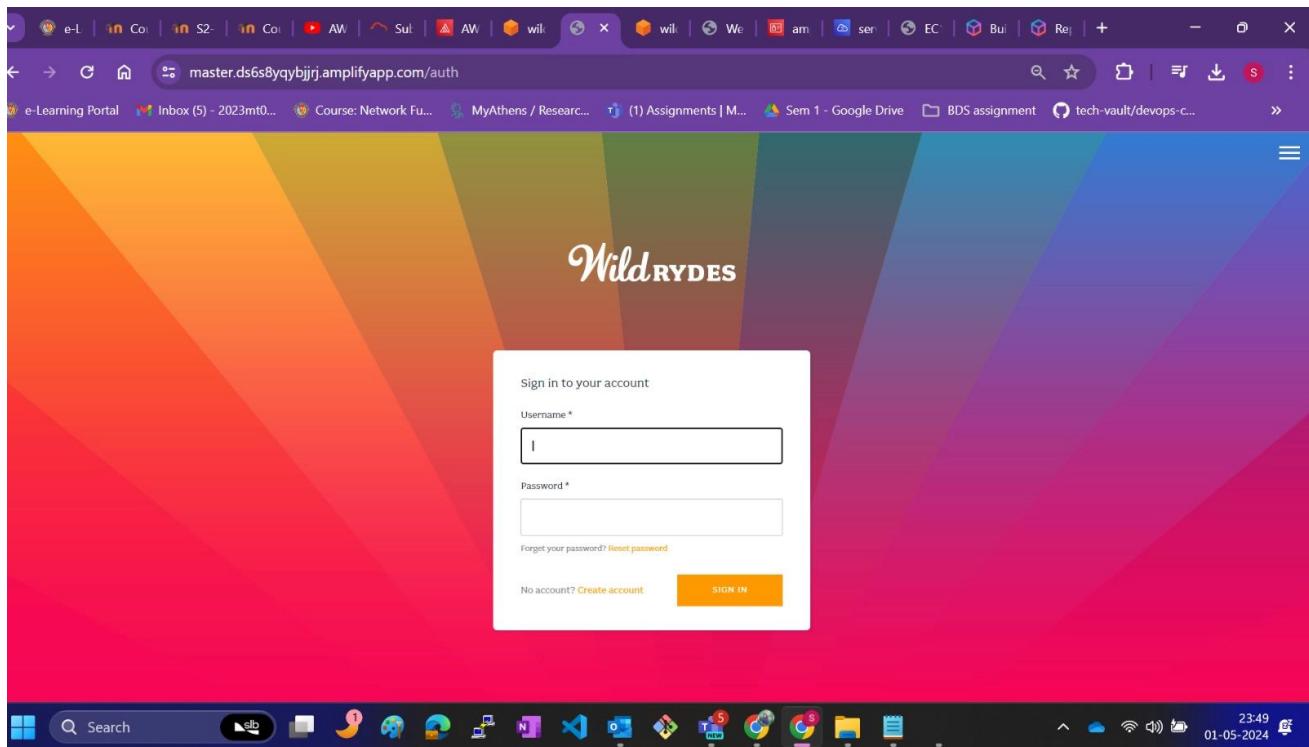
## 27. User pool in cognito created

The screenshot shows the AWS Cognito console with a user pool named "wildrydescf7571d9\_userpool\_cf7571d9-master". The pool was created 8 seconds ago and last updated 8 seconds ago. The status is "Active". A message box at the top right says "New from Amazon Verified Permissions! Cognito user group authorization for API Gateway".

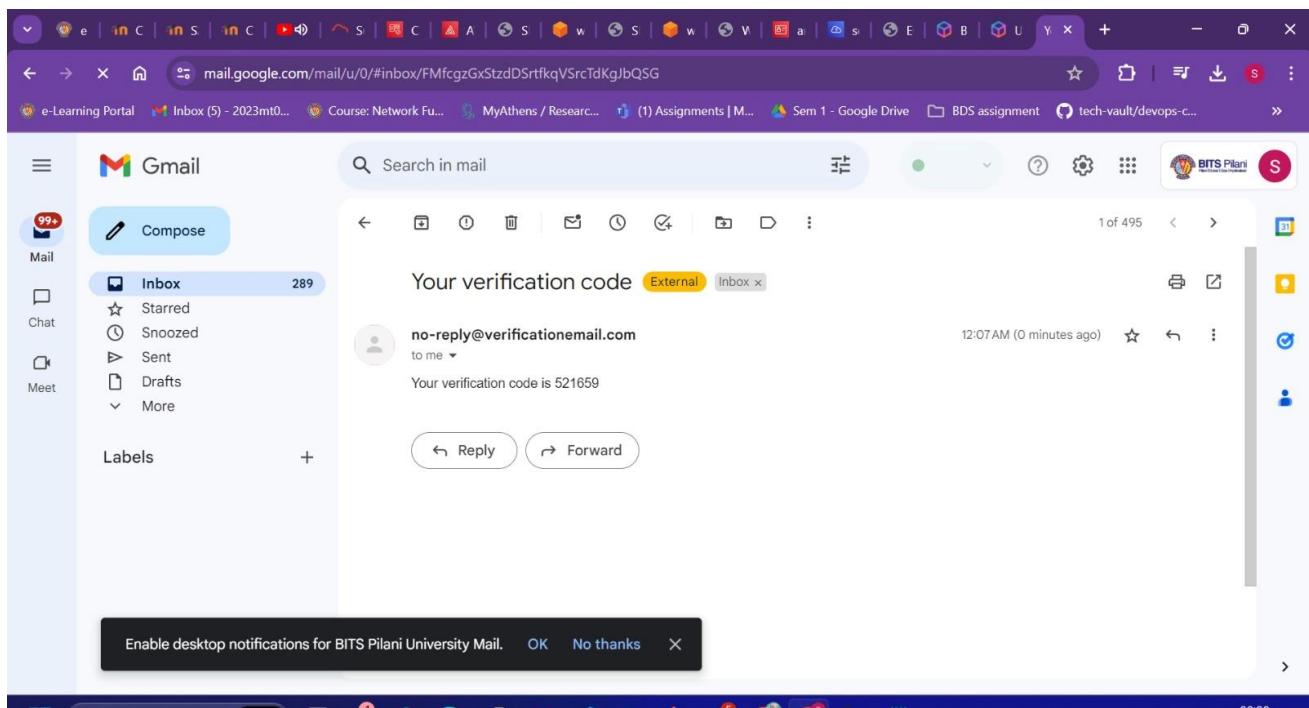
## 28. Deploy

The screenshot shows the AWS Amplify console for the app "wild-rydes". It displays a build step named "Build 9" which has completed successfully. The steps shown are Provision, Build, and Deploy. The build started at 5/2/2024, 12:00:14 AM and took 4 minutes 54 seconds. The domain for the app is https://master.ds6s8yqybrijr.amplifyapp.com.

## 29. New user created



### 30. Verification code received on mail



### 31. New user created on the portal

The screenshot shows the AWS CloudShell interface with the Amazon Cognito console open. The 'Users' tab is selected, showing one user named 'sruthi' with the status 'Unconfirmed' and 'Enabled'. The 'Import users (0)' section below shows no import jobs found.

## 32. Confirming user

The screenshot shows the AWS CloudShell interface with the Amazon Cognito console open. A confirmation dialog box is overlaid on the user details page, asking 'Confirm account for "sruthi"?'. The dialog box contains the text: 'This will confirm the user's account. The user will have immediate access to sign in to your user pool. Learn more'. There are 'Cancel' and 'Confirm' buttons. The background shows the user information and attributes sections.

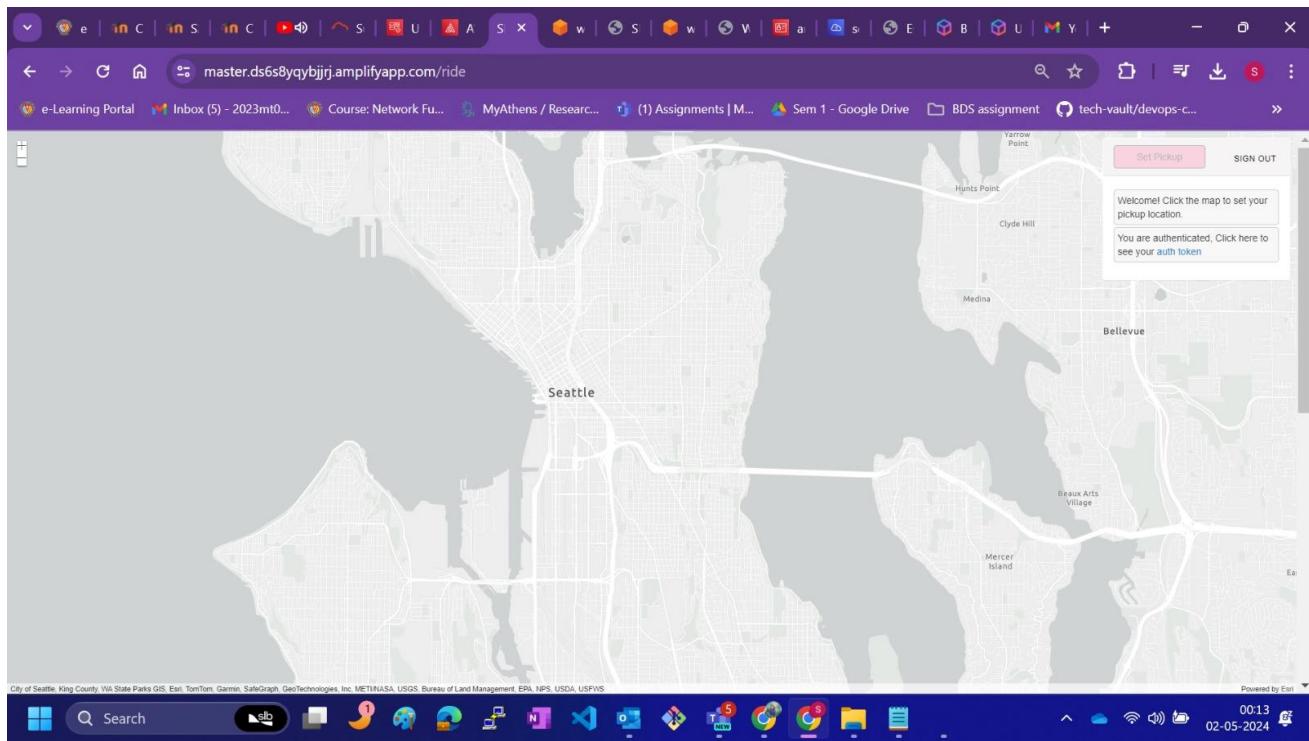
## 33. User confirmed

The screenshot shows the AWS Cognito console interface. At the top, there's a navigation bar with links like 'e-Learning Portal', 'Inbox (5) - 2023mt0...', 'Course: Network Fu...', 'MyAthens / Researc...', '(1) Assignments | M...', 'Sem 1 - Google Drive', 'BDS assignment', and 'tech-vault/devops-c...'. Below the navigation bar, the main content area has tabs for 'User pools' and 'Identity pools'. The 'User pools' tab is selected. Under 'User pools', there's a sub-tab for 'Users'. The 'Users' tab is also selected. The main content area displays a table for 'Users (1) Info'. The table has columns for 'User name', 'Email address', 'Email verified', 'Confirmation status', and 'Status'. There is one row for a user named 'sruthi' with the email '2023mt03005@wilp.bits-pilani.ac.in'. The 'Confirmation status' is 'Confirmed' and the 'Status' is 'Enabled'. Below the table, there's a section for 'Import users (0) Info' with a table showing 'Imported users', 'Skipped users', 'Failed users', and 'CloudWatch logs'. A 'Create import job' button is available. At the bottom of the page, there's a footer with links for 'CloudShell', 'Feedback', and 'Cookie preferences', along with system status icons.

### 34. Signed in

The screenshot shows a web browser window with the URL 'master.ds6s8yqybijrj.amplifyapp.com/ride'. The browser's address bar also shows 'e-Learning Portal', 'Inbox (5) - 2023mt0...', 'Course: Network Fu...', 'MyAthens / Researc...', '(1) Assignments | M...', 'Sem 1 - Google Drive', 'BDS assignment', and 'tech-vault/devops-c...'. The main content area displays a 'YOUR AUTH TOKEN' modal. The modal contains a message: 'This page is not functional yet because there is no API invoke URL configured in /src/config.js. You'll configure this in Module 1.' It also says, 'In the meantime, if you'd like to test the Amazon Cognito user pool authorizer for your API, use the auth token below.' Below this message is a large text box containing a long auth token: 'eyJraWQiOiN0WtyaGFG0MTE0Ri3TnB0dkE0a3U4UTBTWyoNWRwWHMs5eU9JS mcoQik0PSislnfS2yl6iUTMjJ2in0 eyJzdWIiOiIxNDE4YzRiOC1jMGIxLTcwODA1NmQ'. At the bottom of the modal is a 'Close' button. Above the modal, there are two buttons: 'Set Pickup' and 'SIGN OUT'. To the right of the modal, there's a message: 'Welcome! Click the map to set your pickup location.' and another message: 'You are authenticated. Click here to see your auth token.' At the bottom of the page, there's a footer with links for 'CloudShell', 'Feedback', and 'Cookie preferences', along with system status icons.

### 35. Website



## 36. Creating a table in DynamoDB

The screenshot shows the AWS DynamoDB console. The 'Tables' section lists a single table named 'Rides' with status 'Active'. The table has a partition key 'Rides' and a sort key '-'. It uses 'Provisioned' read and write capacity modes with a capacity of 5. The table is 0 bytes in size and is of the 'Standard' type. A success message at the top says "The Rides table was created successfully."

## 37. Creating IAM role

**Select trusted entity**

**Trusted entity type**

- AWS service
- AWS account
- Web identity
- SAML 2.0 federation
- Custom trust policy

**Use case**

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

Service or use case

Choose a service or use case

Cancel Next

### 38. New role name

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**WildRydesLambda**

**Description**  
Add a short explanation for this role.  
Allows Lambda functions to call AWS services on your behalf.

**Step 1: Select trusted entities**

**Trust policy**

```

1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-     }
10-  ]
11-}
  
```

Edit

Next Step

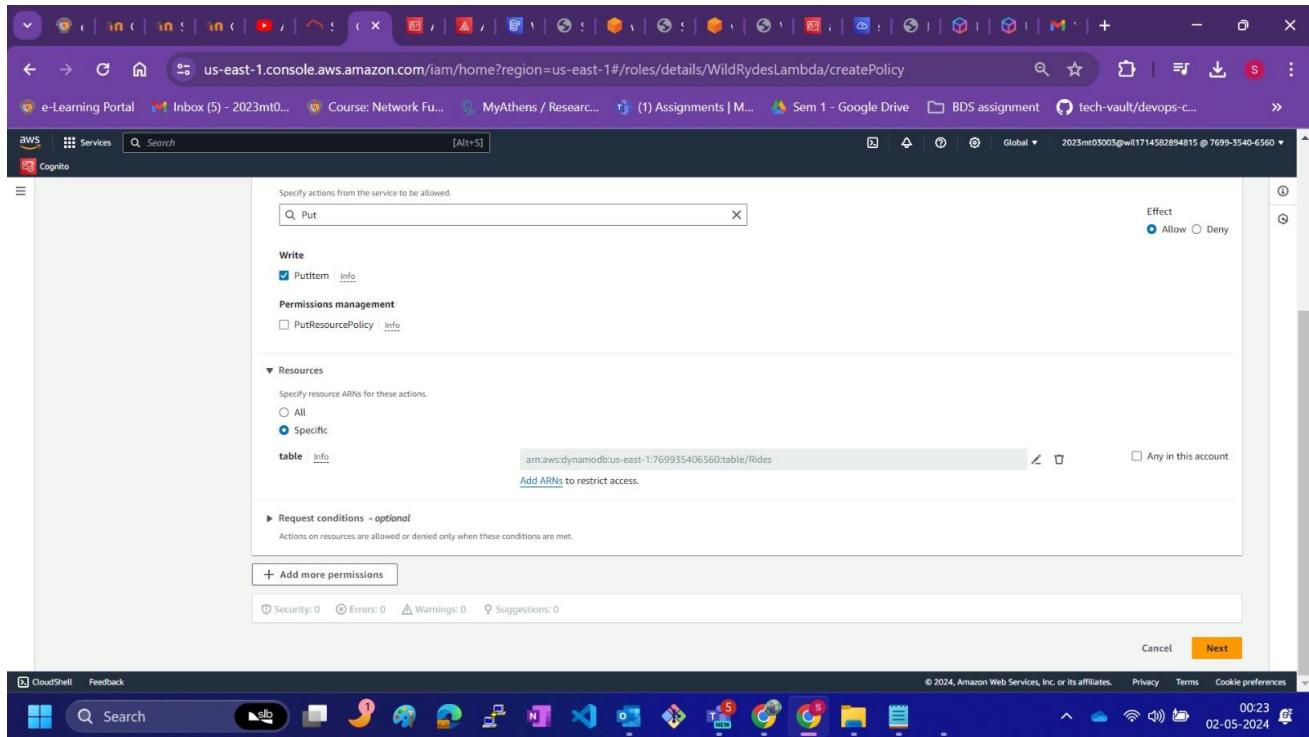
### 39. New role with permission

The screenshot shows the AWS IAM console with the 'WildRydesLambda' role selected. The ARN of the role is displayed as `arn:aws:iam::769935406560:role/WildRydesLambda`. The permissions tab shows one managed policy attached: `AWSLambdaBasicExecutionRole`.

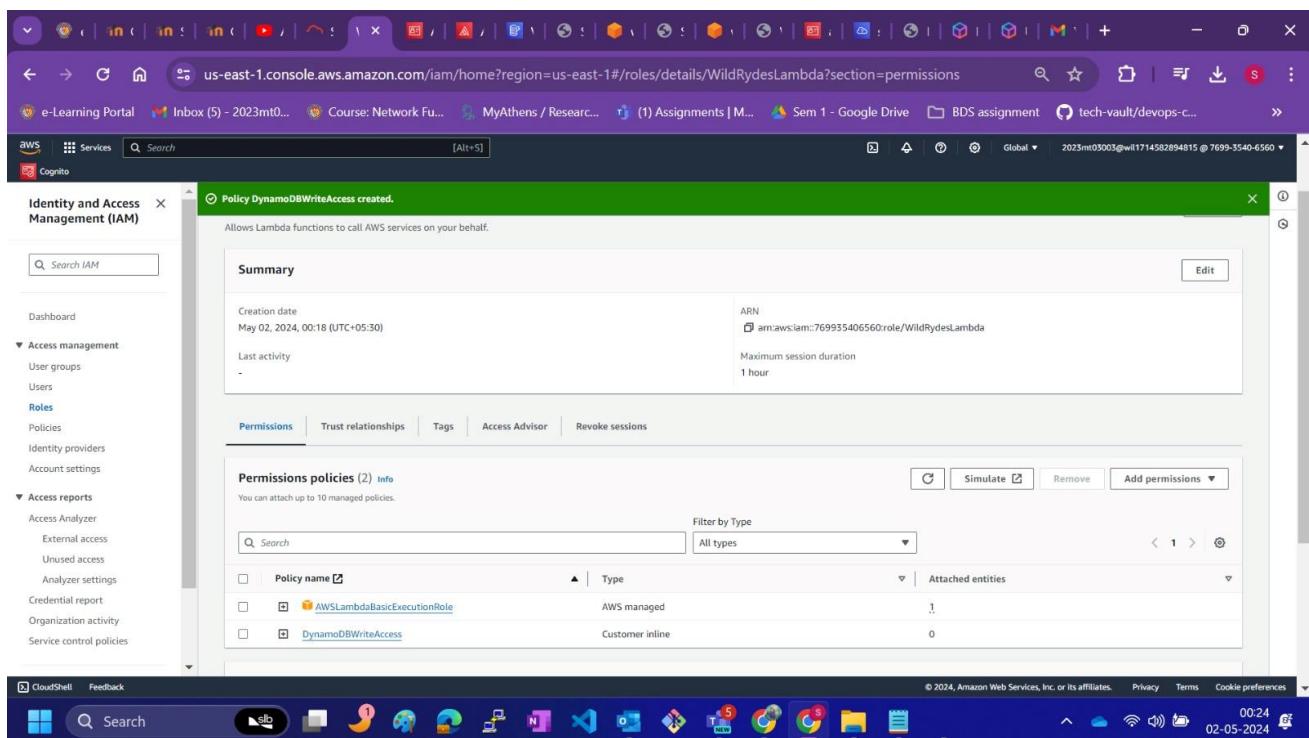
#### 40. Taking the ARN of the DynamoDB table

The screenshot shows the AWS DynamoDB console with the 'Rides' table selected. The ARN of the table is copied to the clipboard, as indicated by the green highlighted text `arn:aws:dynamodb:us-east-1:769935406560:table/Rides`.

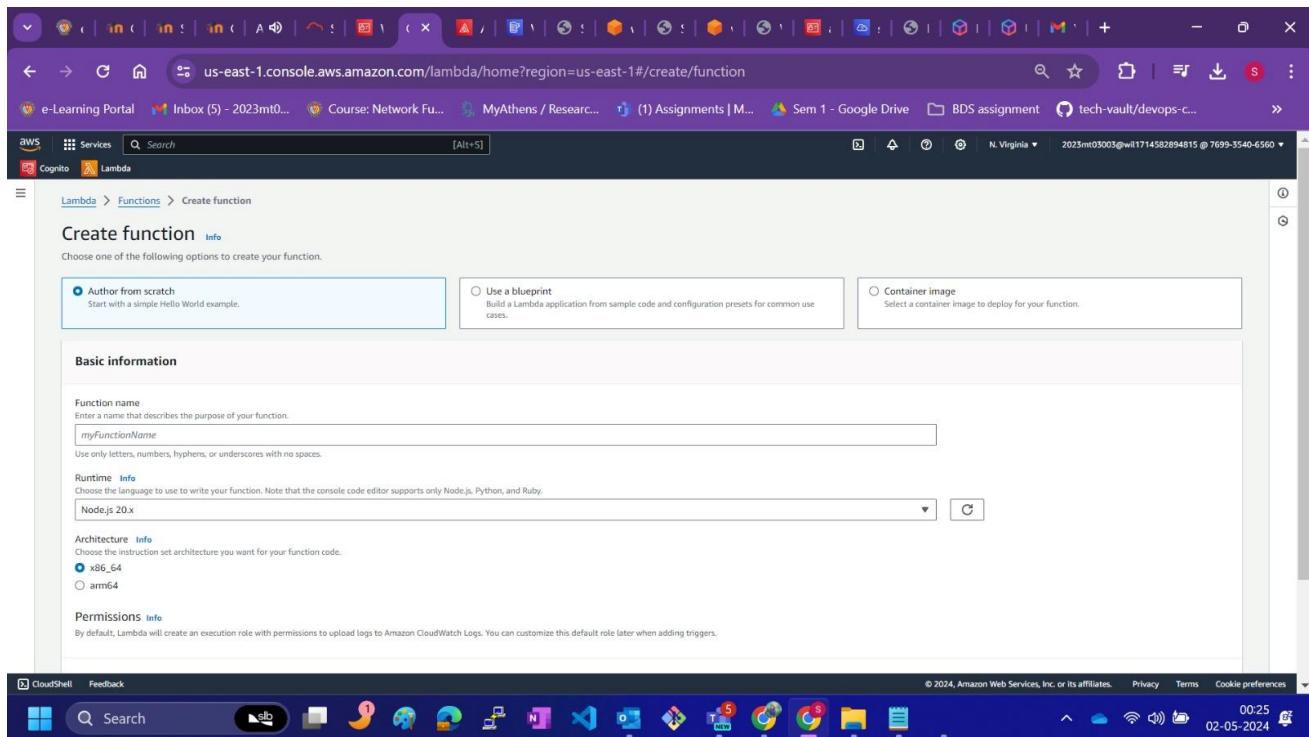
#### 41. Added ARN to the policy



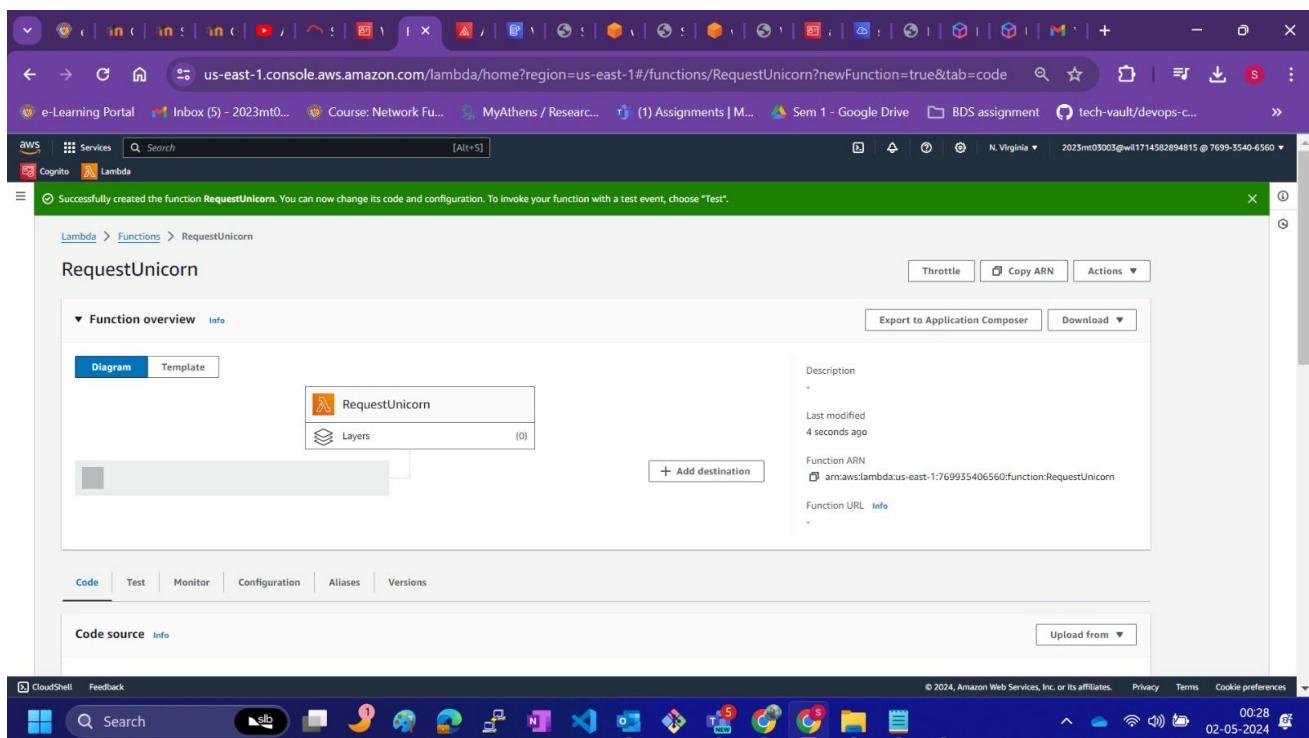
## 42. Created DynamoDB write policy



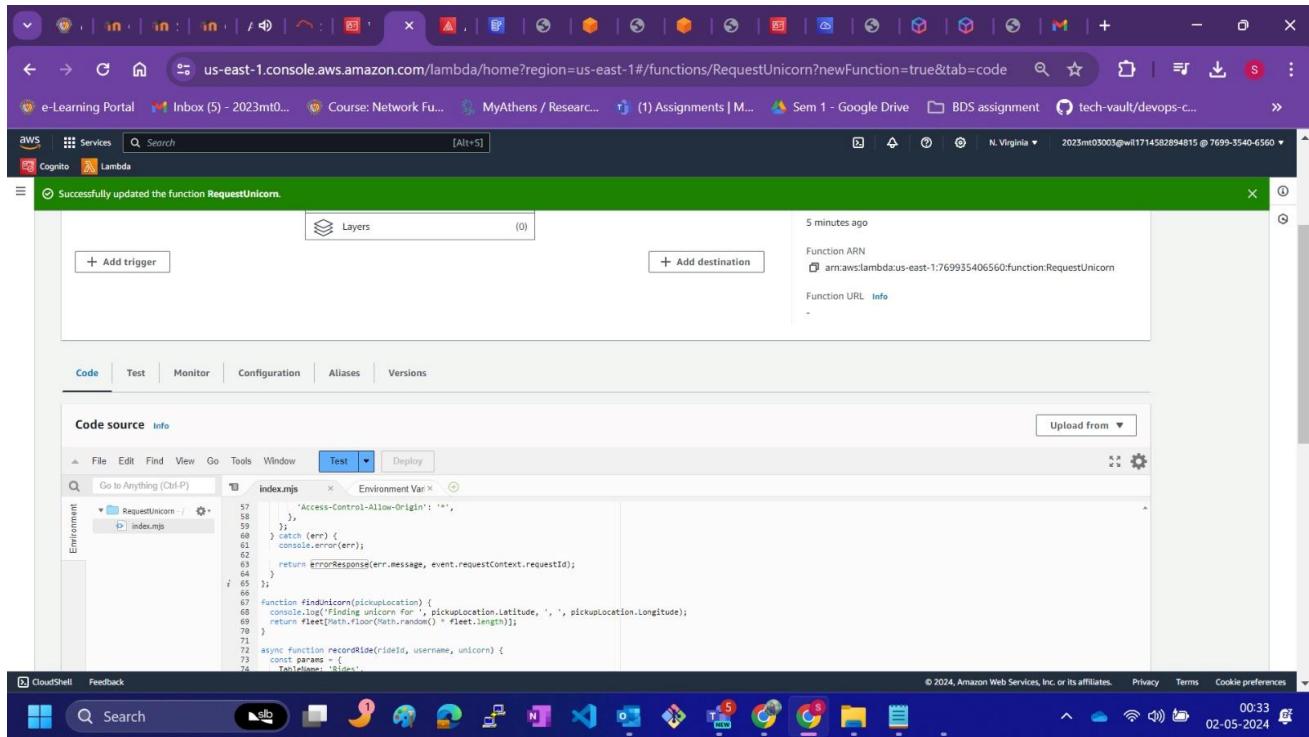
## 43. Creating new Lambda function



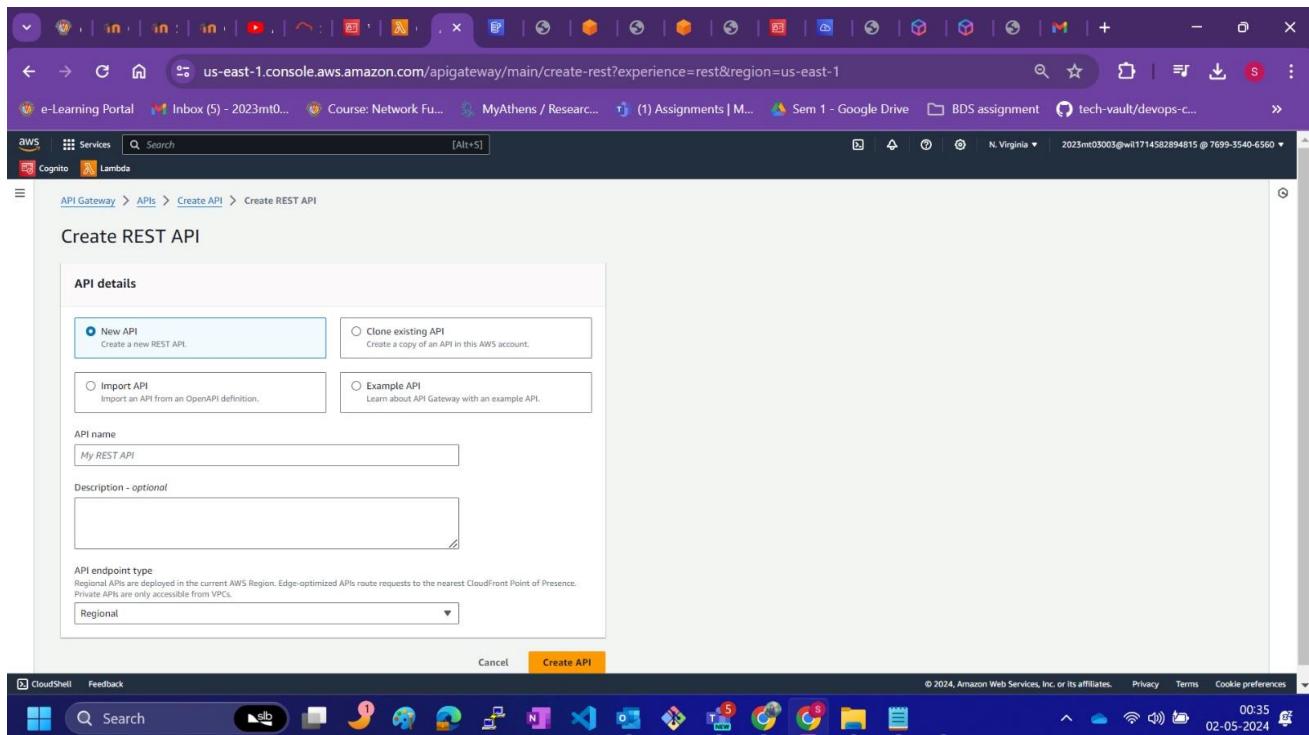
#### 44. Created lambda function



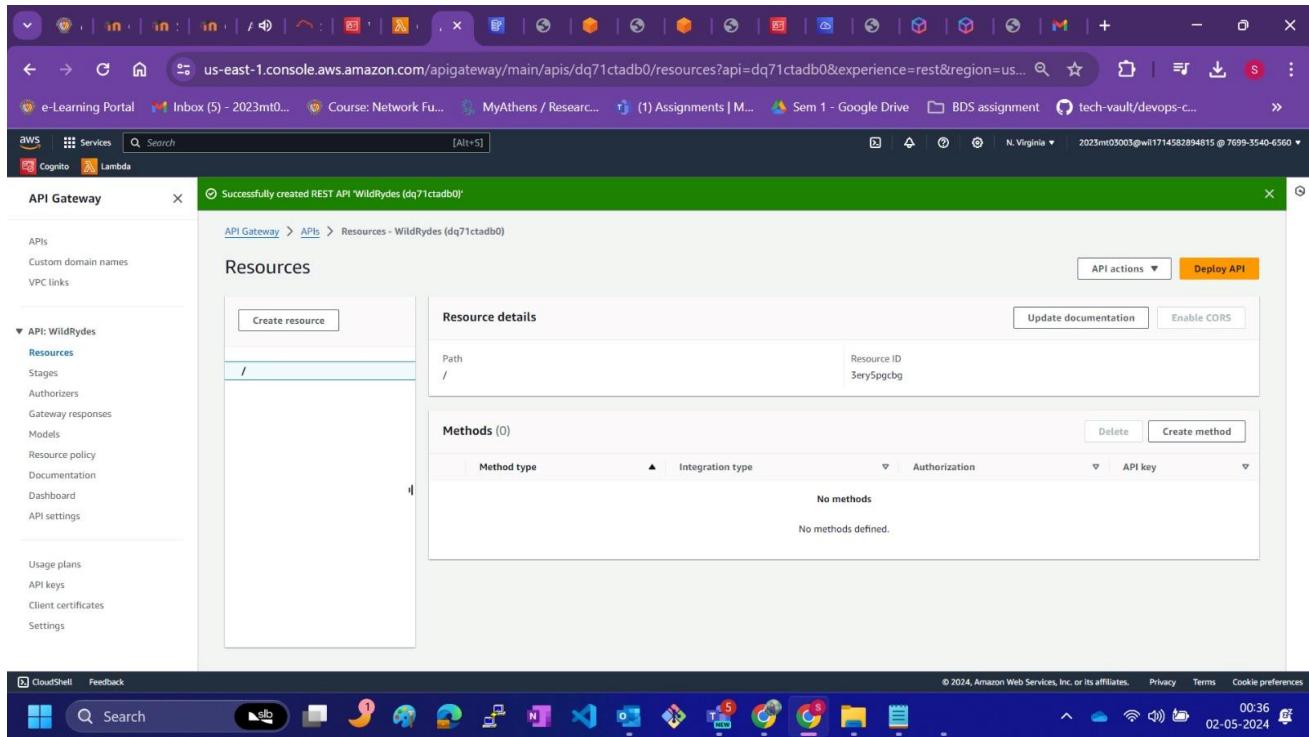
#### 45. Updated index.js



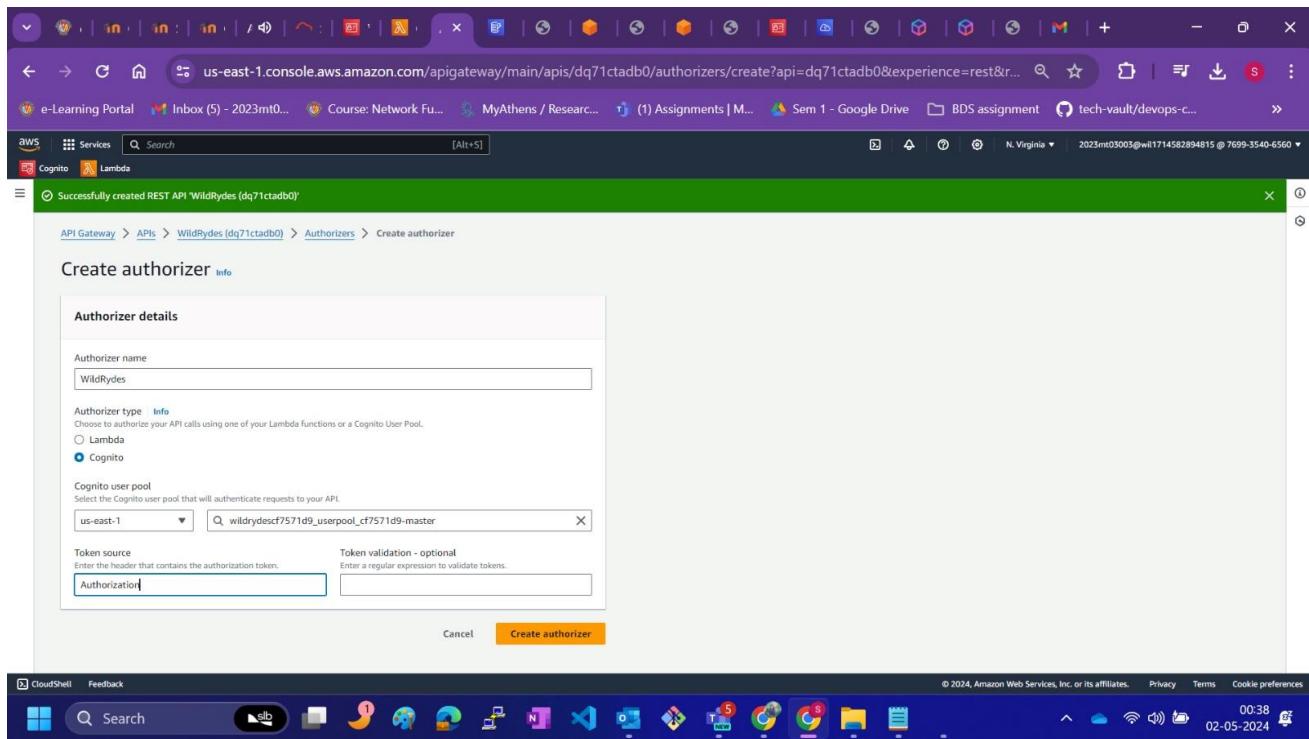
## 46. Creating a REST api



## 47. Created the api



## 48. Creating authorizer



## 49. Pasting token from the website

The screenshot shows the AWS API Gateway Authorizers page. A green success message at the top says "Successfully created authorizer 'WildRydes'". The "Authorizer details" section shows the Authorizer ID as "mnrrpakk" and the Token source as "Authorization". The "Test authorizer" section contains a simulated invocation request with a token value: "eyJraWQiOiJONWtyaGF0MTE0Rit3TnB0dkE0a3U4UTBTWWcyNWRwWh".

## 50. Testing out authorizer. 200 success code

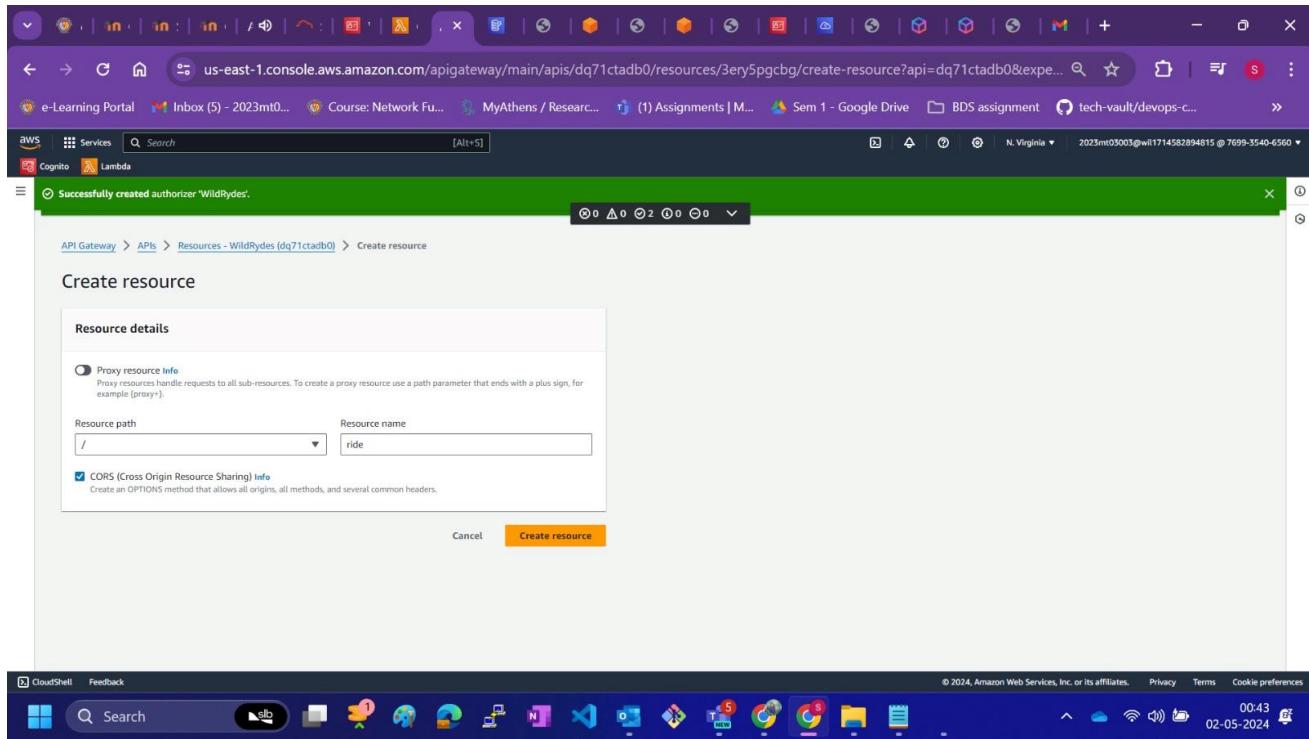
The screenshot shows the AWS API Gateway Test authorizer page. A yellow button labeled "Test authorizer" is visible. The "Token value" field contains the same Cognito JWT as before. The "Test authorizer" section displays a successful response with a status code of 200 and a claims object containing various user information.

```

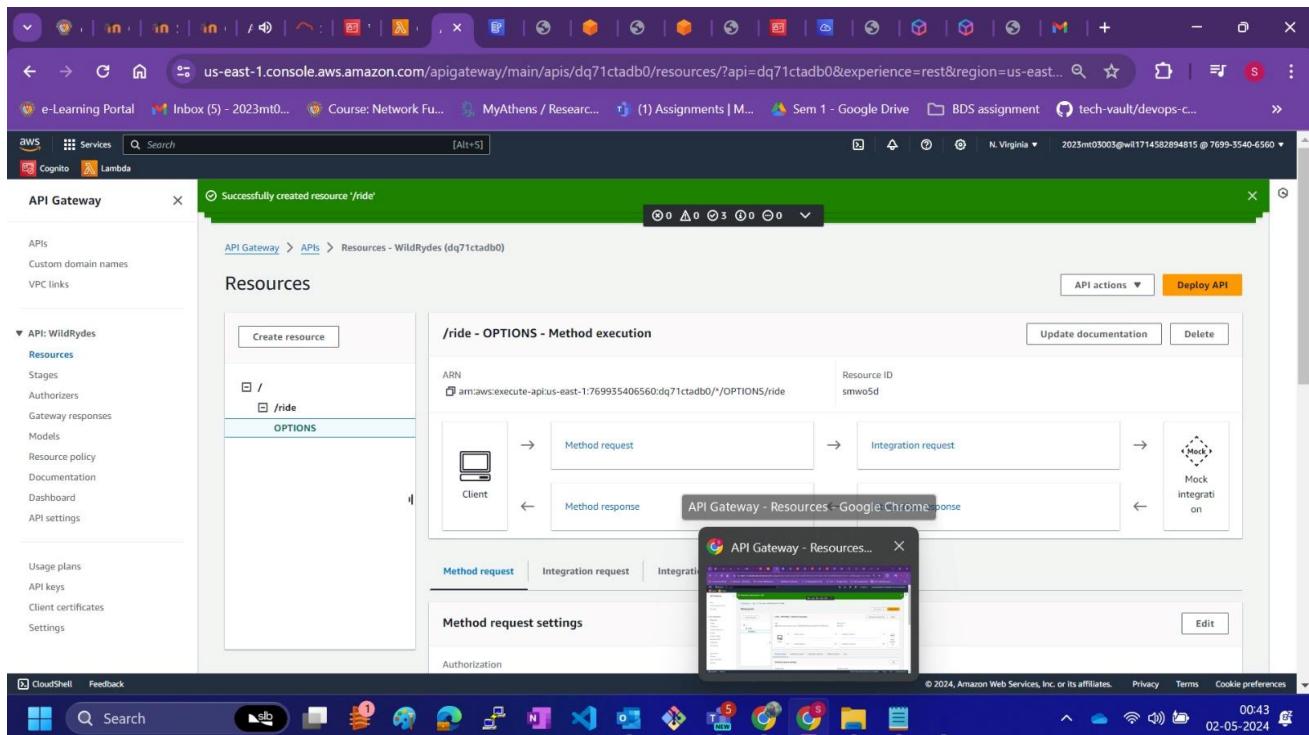
{
  "aud": "121388rmn426hsaa8c5d49d4a5",
  "auth_time": "1714588850",
  "cognito:username": "sruthi",
  "email": "2023mt03003@ulip.bits-pilani.ac.in",
  "email_verified": "false",
  "event_id": "28376fc9-ee02-46af-89df-015eaa0f35cc",
  "exp": "Wed May 01 10:42:33 UTC 2024",
  "iat": "Wed May 01 18:42:33 UTC 2024",
  "iss": "https://cognito-idp.us-east-1.amazonaws.com/us-east-1_uqQPVs4KF",
  "jti": "96cd1381-5971-44c9-9678-e7135b7f63b9",
  "origin_jti": "4f751b63-ebda-48a8-9f24-e2633e00404c",
  "phone_number": "+919433656197",
  "phone_number_verified": "false",
  "sub": "1418c4e8-c0b1-7080-6d12-002116cf89e6",
  "token_use": "id"
}

```

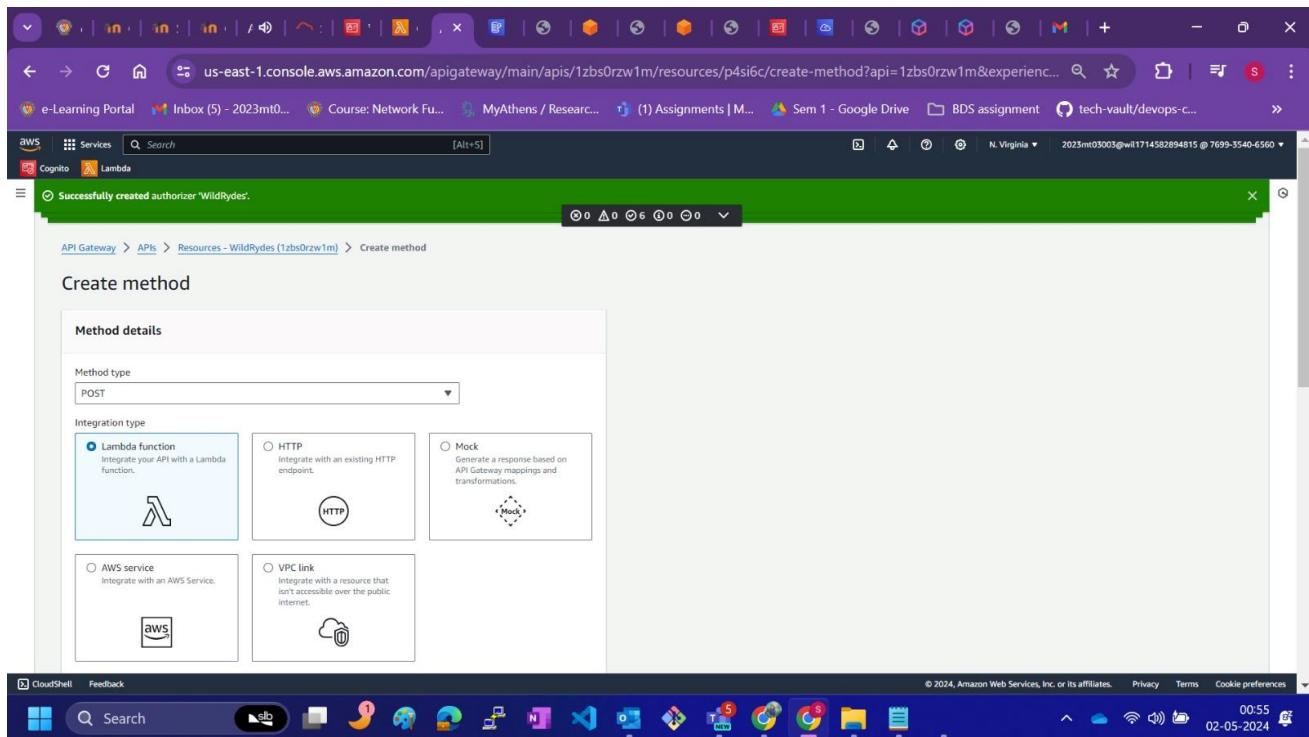
## 51. Creating resource webhook with /ride



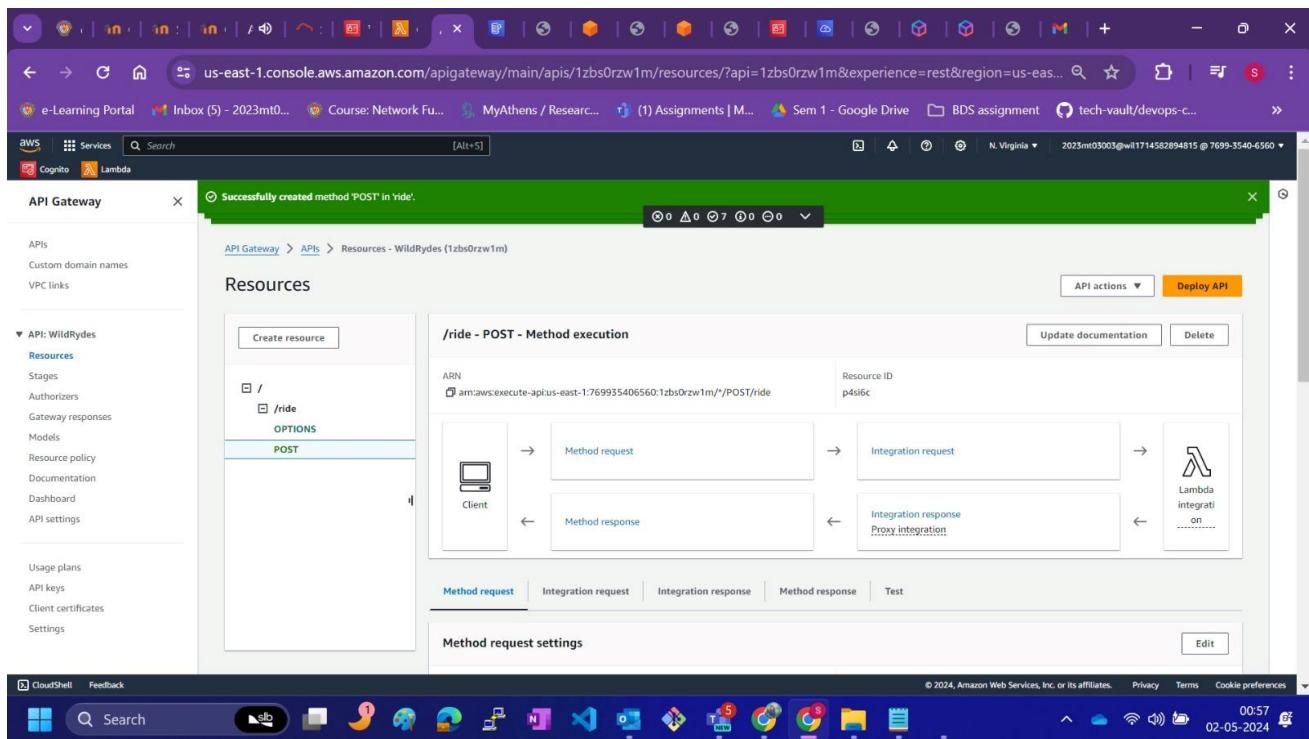
52. Creation successful



53. Creating POST method



54. POST method created



55. Deployed API

The screenshot shows the AWS API Gateway console. A green banner at the top indicates a successful deployment for the WildRydes API. The left sidebar shows the API structure: API: WildRydes, Resources, Stages (selected), Authorizers, Gateway responses, Models, Resource policy, Documentation, Dashboard, API settings, Usage plans, API keys, Client certificates, and Settings. The main pane displays the Stages section, specifically the prod stage. Stage details include:

- Stage name: prod
- Cache cluster info: Inactive
- Default method-level caching: Inactive
- Invoke URL: https://1zbs0rzw1m.execute-api.us-east-1.amazonaws.com/prod
- Active deployment: tr29km on May 02, 2024, 00:58 (UTC+05:30)

Below the stage details is a Logs and tracing section.

## 56. Pasting api url in cloud 9

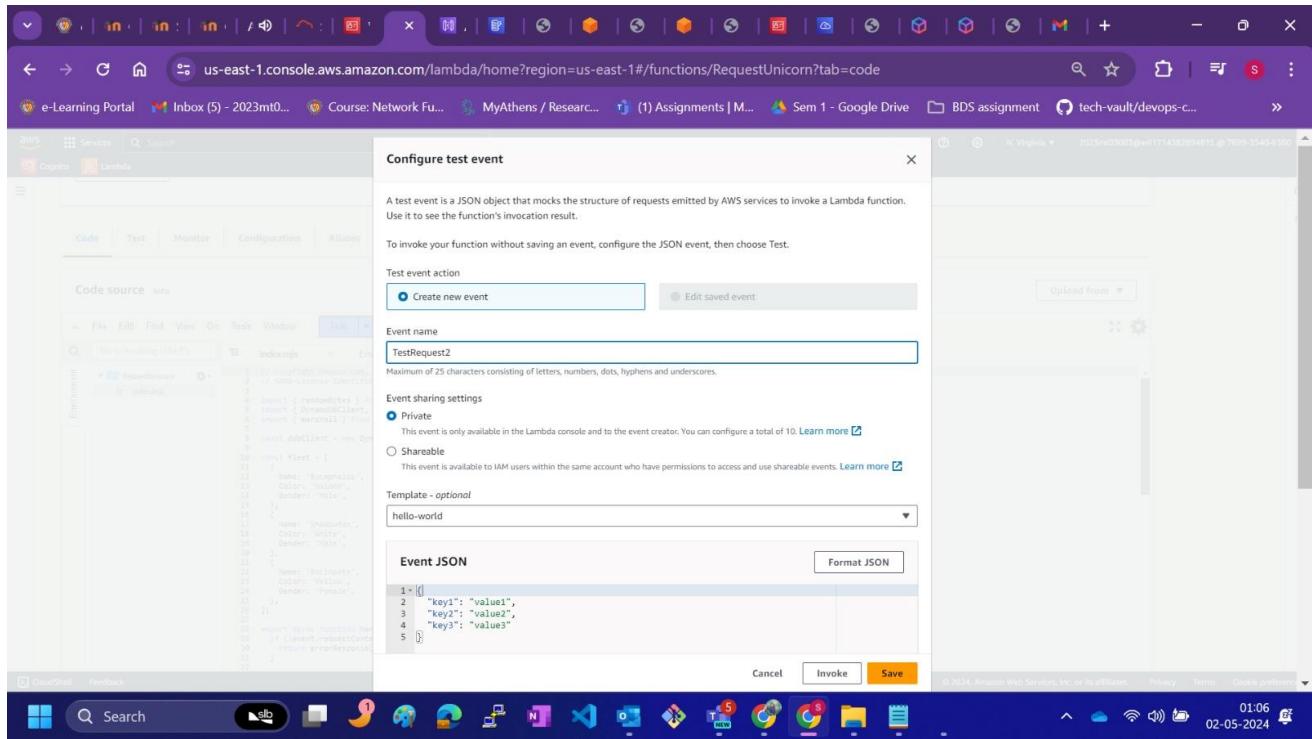
The screenshot shows the Cloud9 IDE interface. The code editor has three tabs: index.html, package.json, and config.js. The config.js file contains the following code:

```
// Add your API invoke URL below, configured in Module 3.
module.exports = {
  api: {
    invokeUrl: "https://1zbs0rzw1m.execute-api.us-east-1.amazonaws.com/prod"
  }
}
```

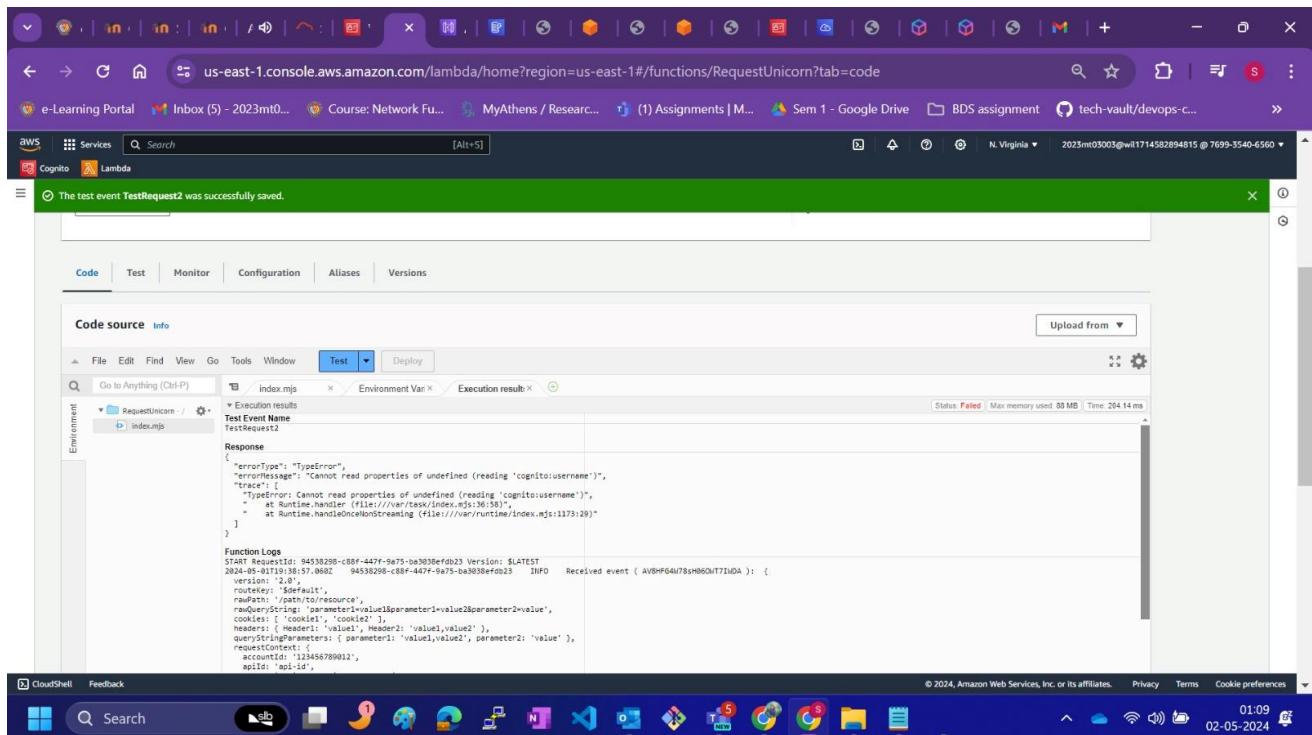
The terminal window at the bottom shows a command being run:

```
bash -c "git add . & git commit -m 'Initial commit' & git push -u origin master"
Comitting objects: 100% (15/15), done.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 3.44 KB | 503.00 KB/s, done.
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
remote: Total 18 (delta 2), reused 0 (delta 0)
To https://git.heroku.com/wild-rydes.git
   2e0f8be..cb6b6df  master -> master
2023-05-02T09:59:45.000Z [will1714582894815]@environment:wild-rydes (master) $
```

## 57. Creating API tests



## 58. API test



## 59. Accessing through API

