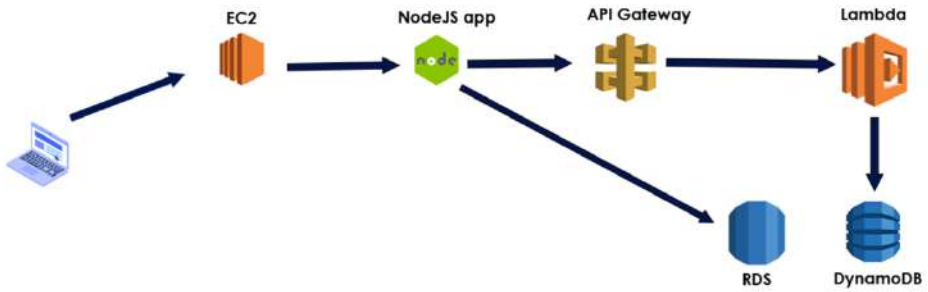
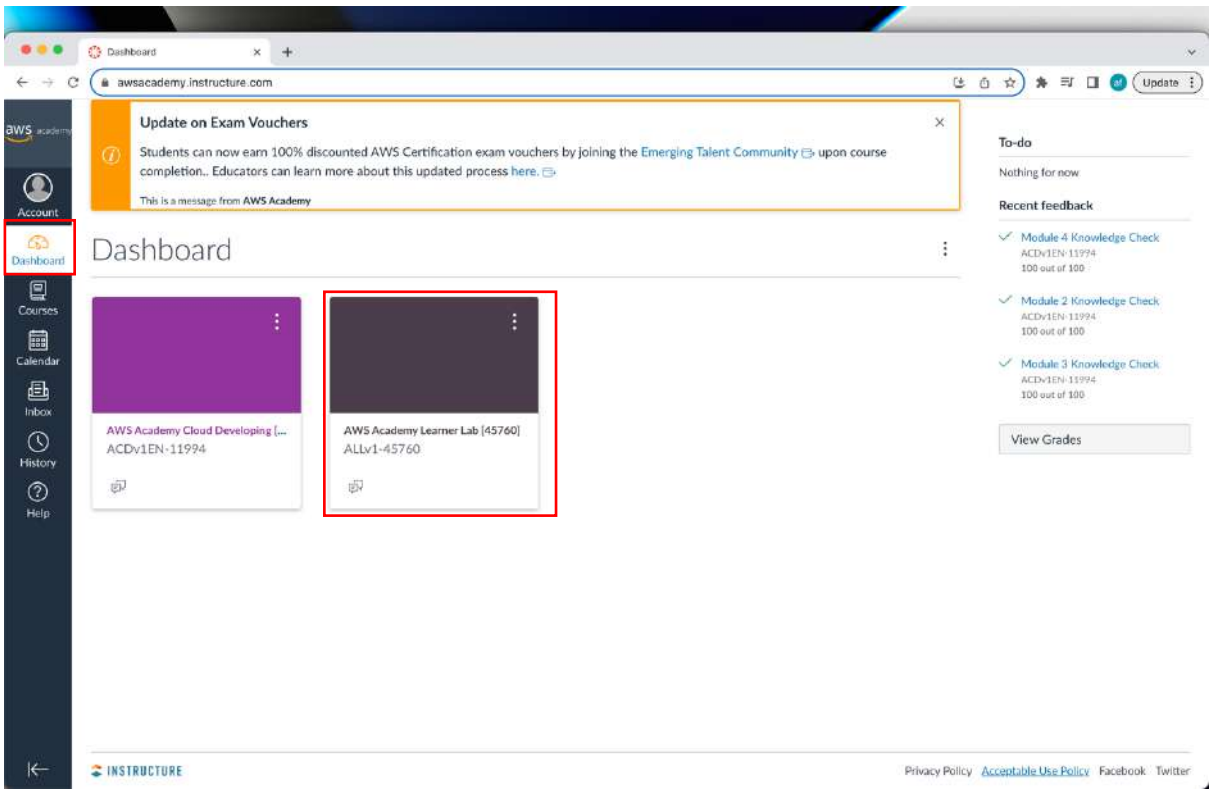
1. Architecture Diagram



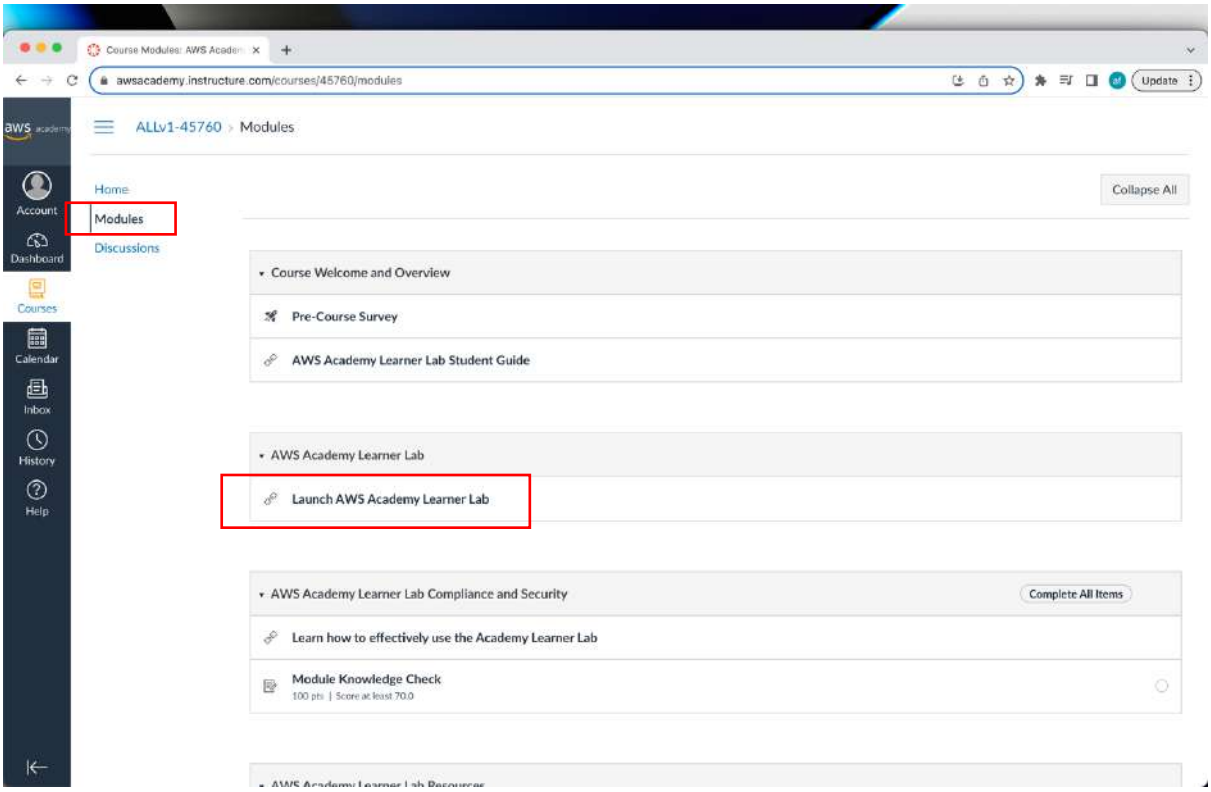
2. Creating an EC2 instance

- Login into AWS academy account.

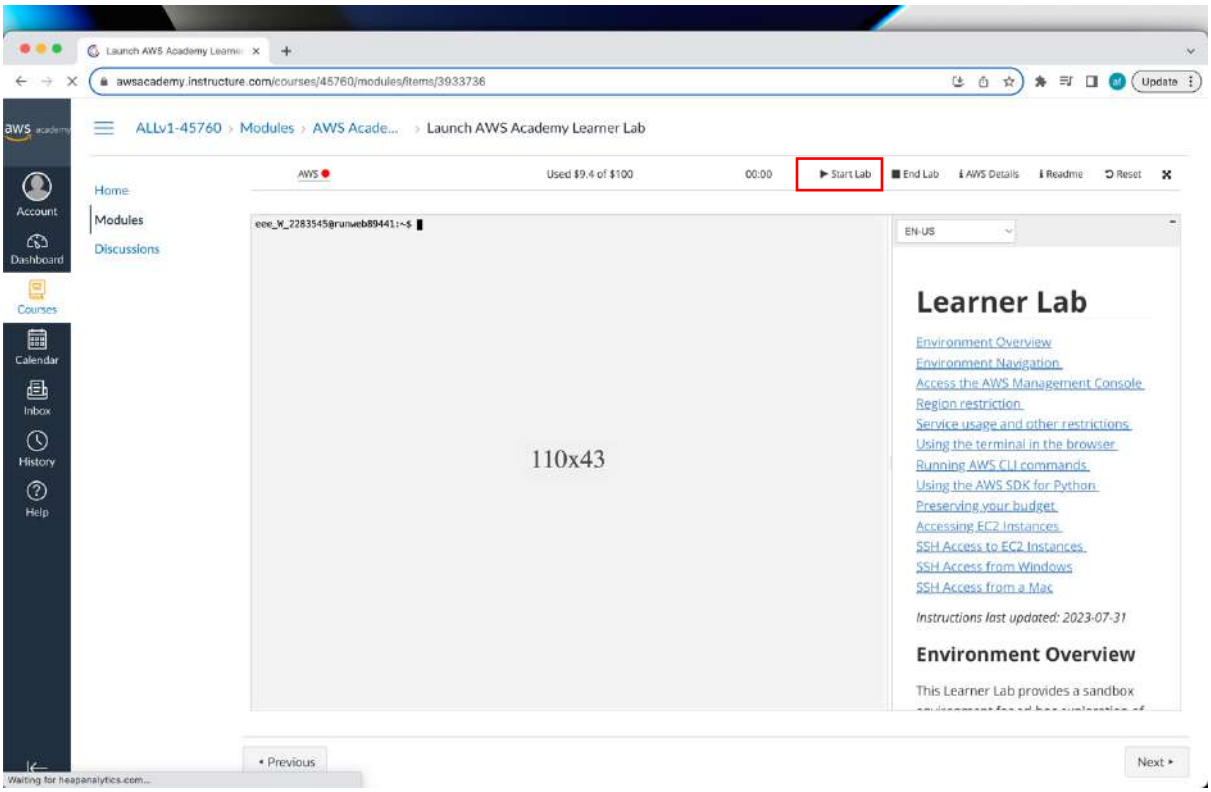
- Navigate to AWS Academy Learner Lab [45760] via the dashboard.



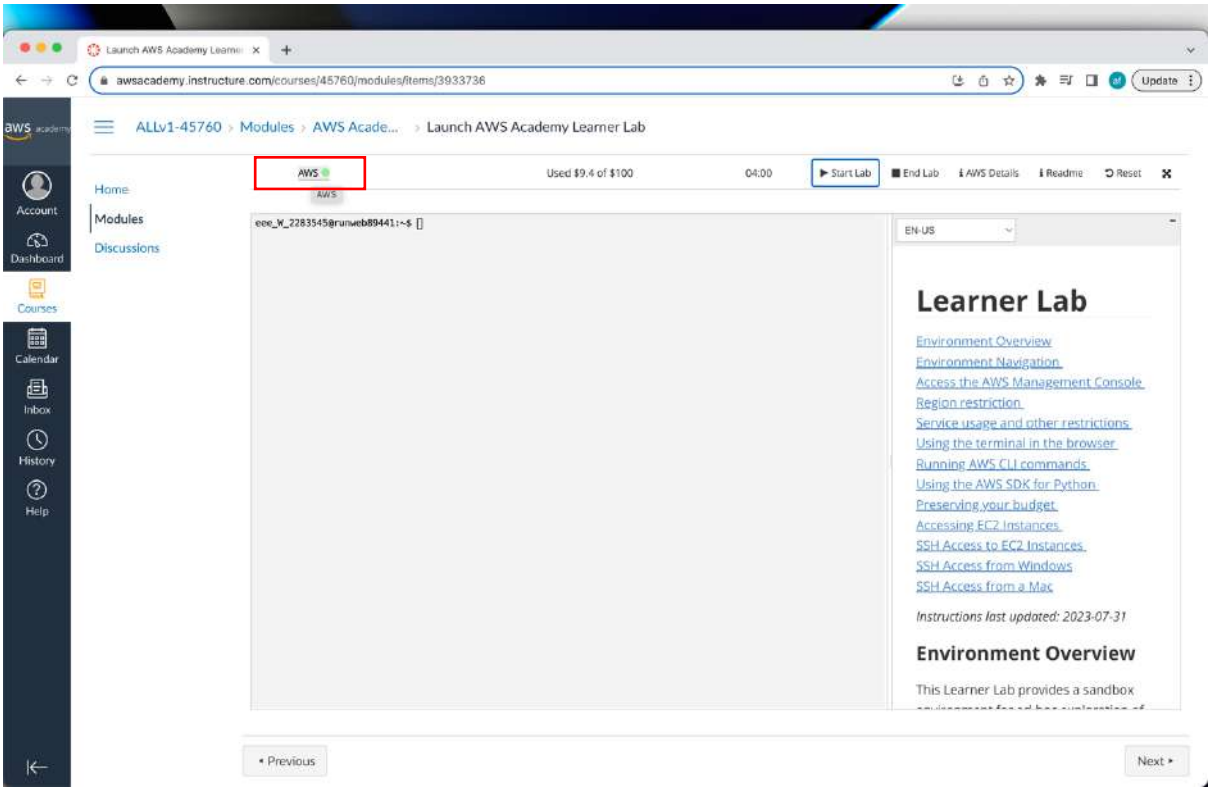
- Click on Modules section and launch the AWS Academy Learner Lab.



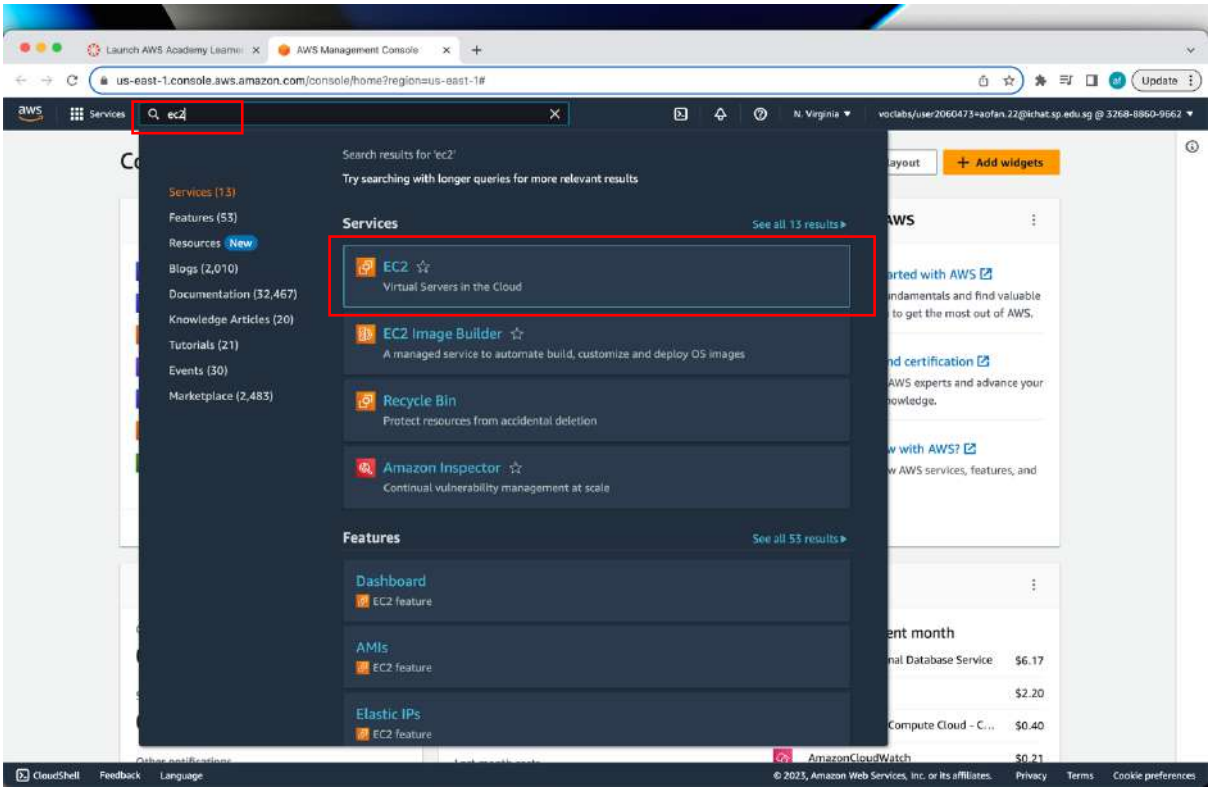
- Start the Lab.



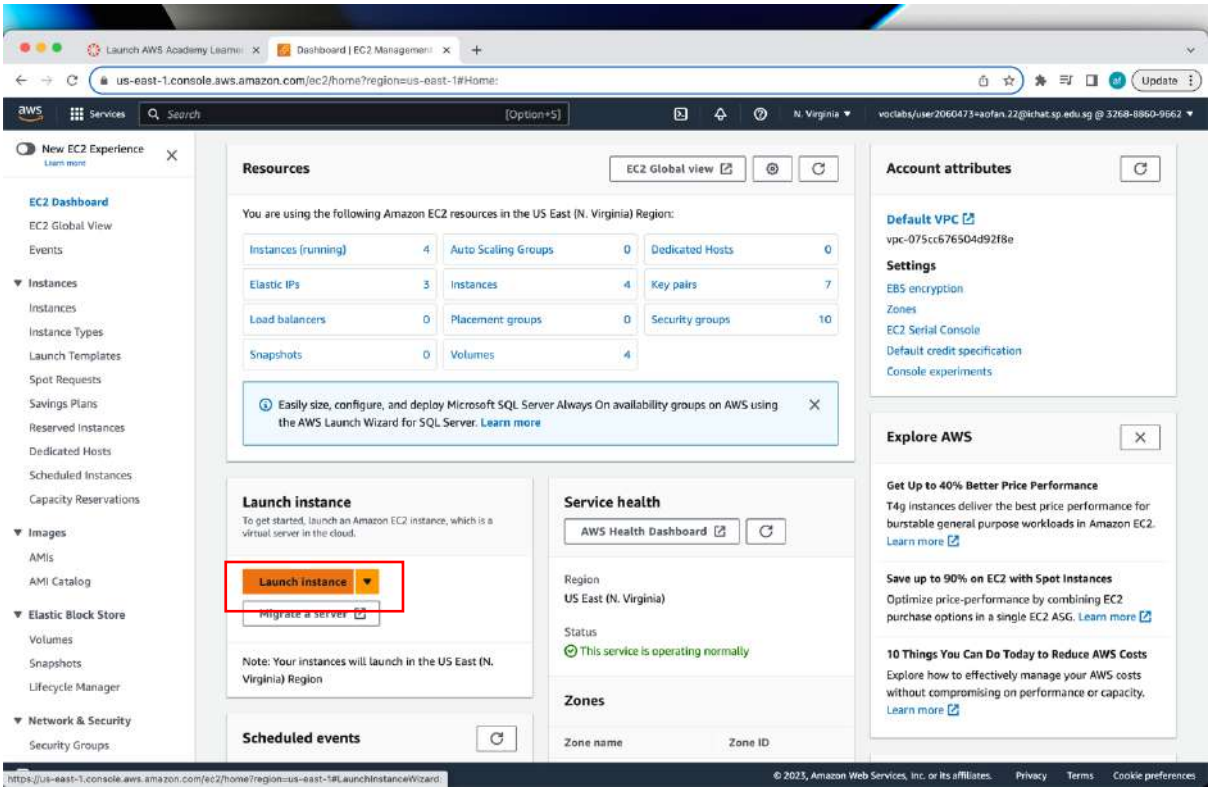
- Click on AWS to navigate to the AWS Console.



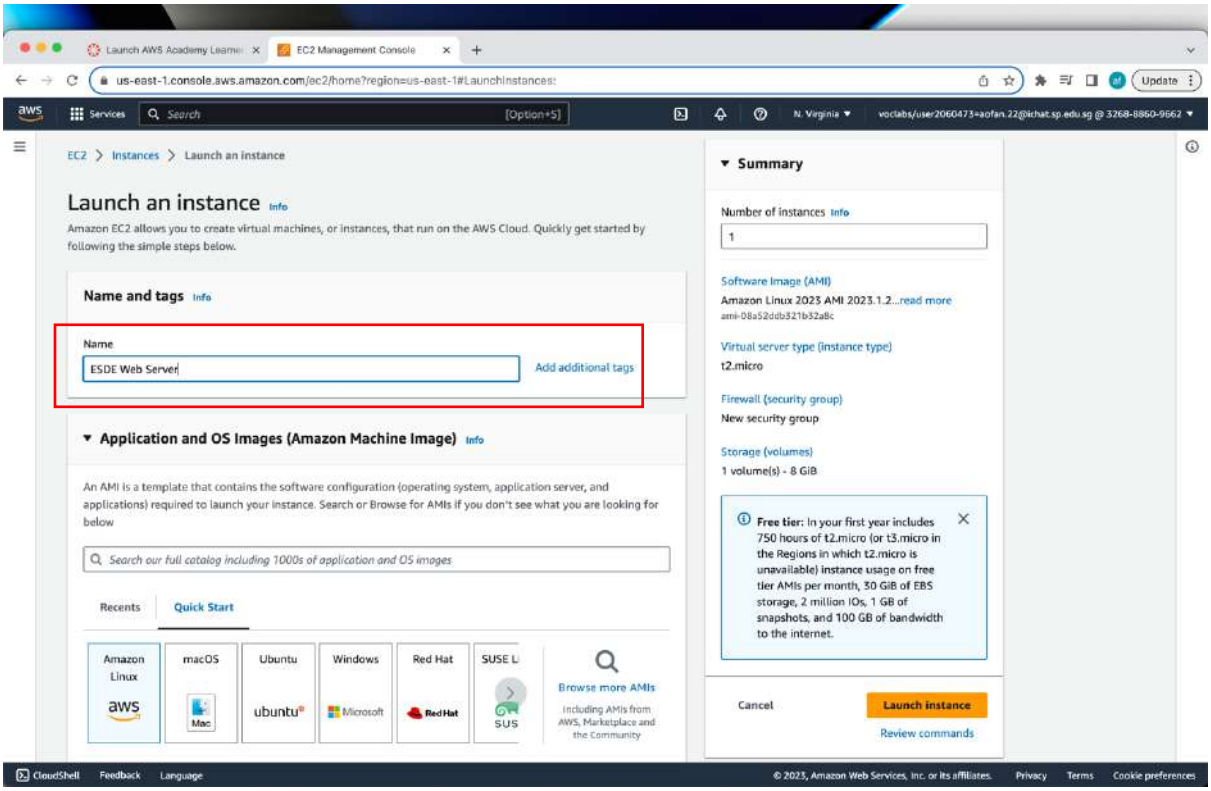
- Search “EC2” in AWS Console to go to EC2 Dashboard.



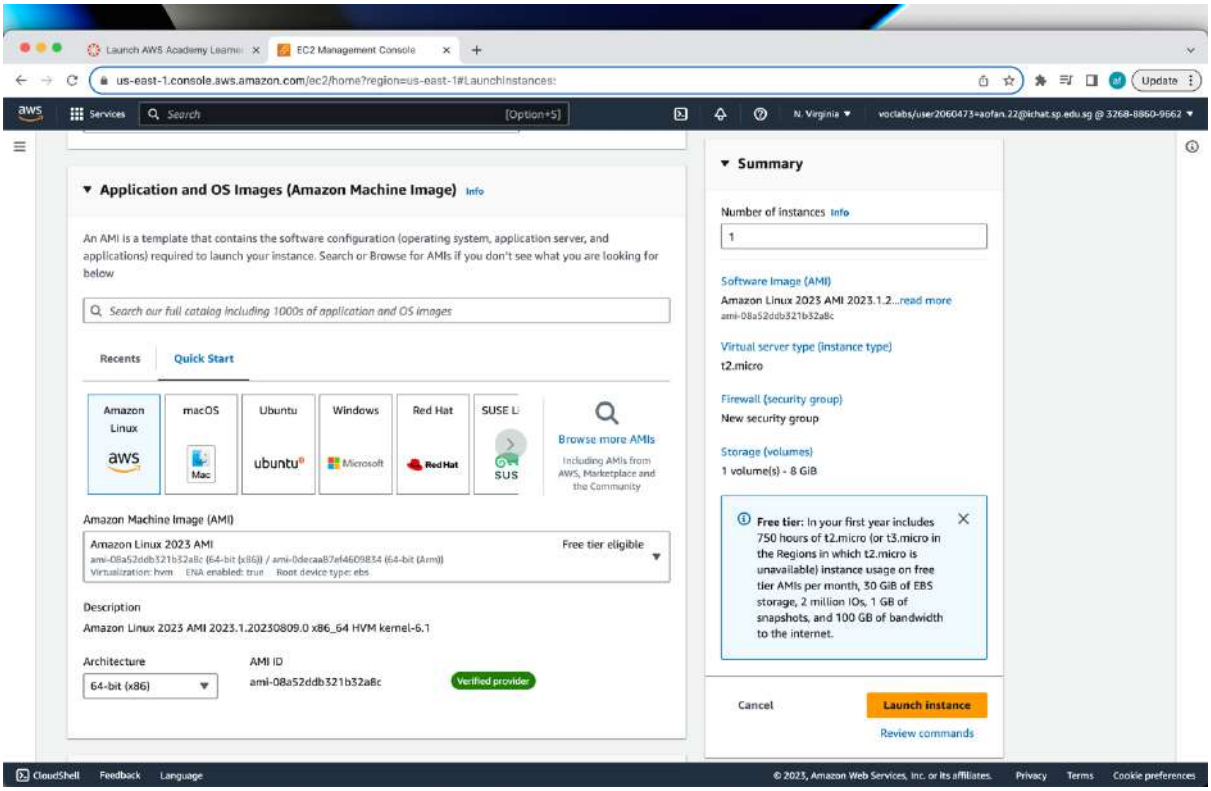
- Click on Launch instance to create a new EC2 instance.



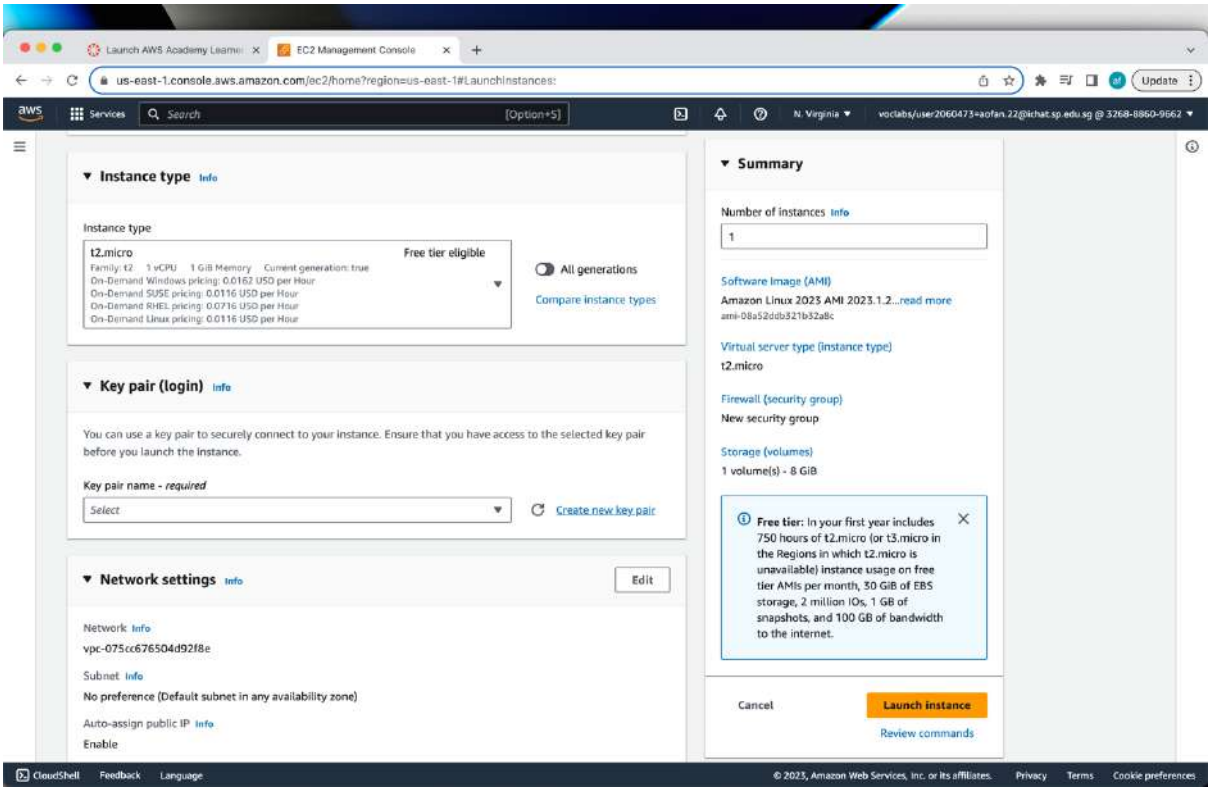
- For Name and tags: ESDE Web Server.



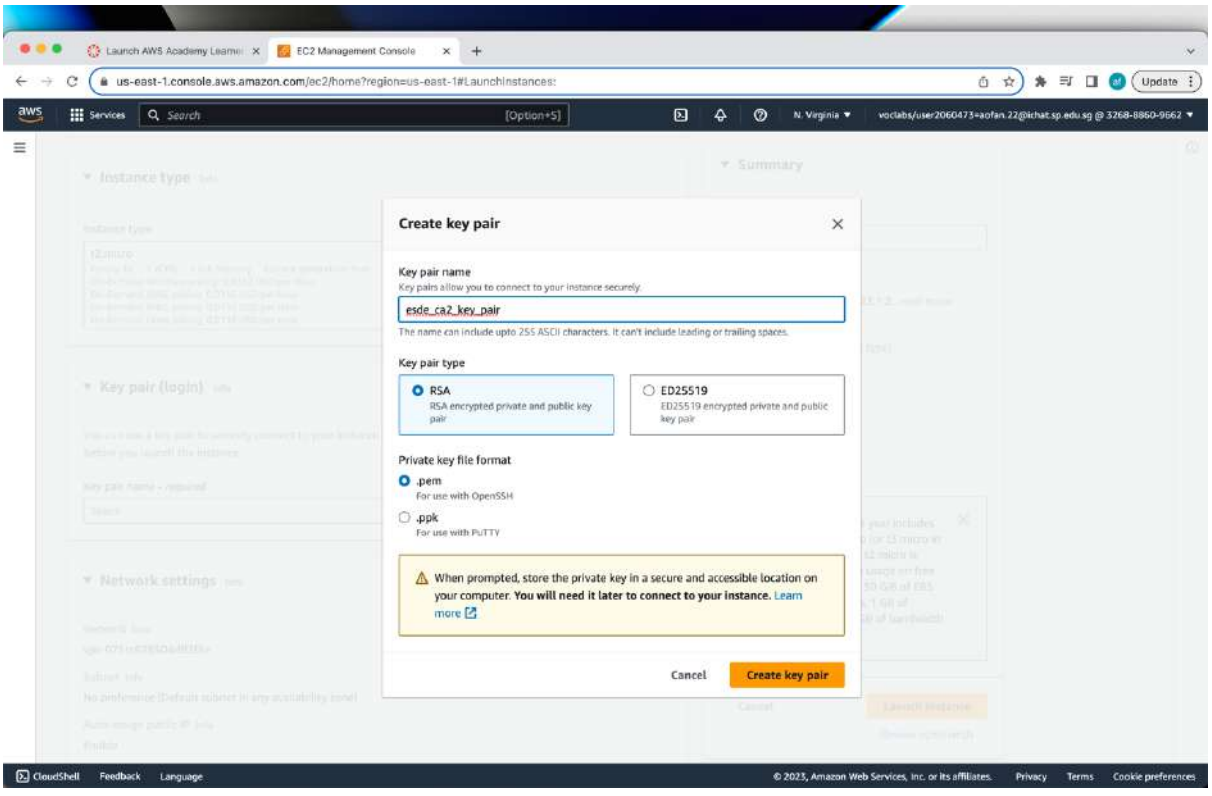
- For Application and OS Images (Amazon Machine Image):



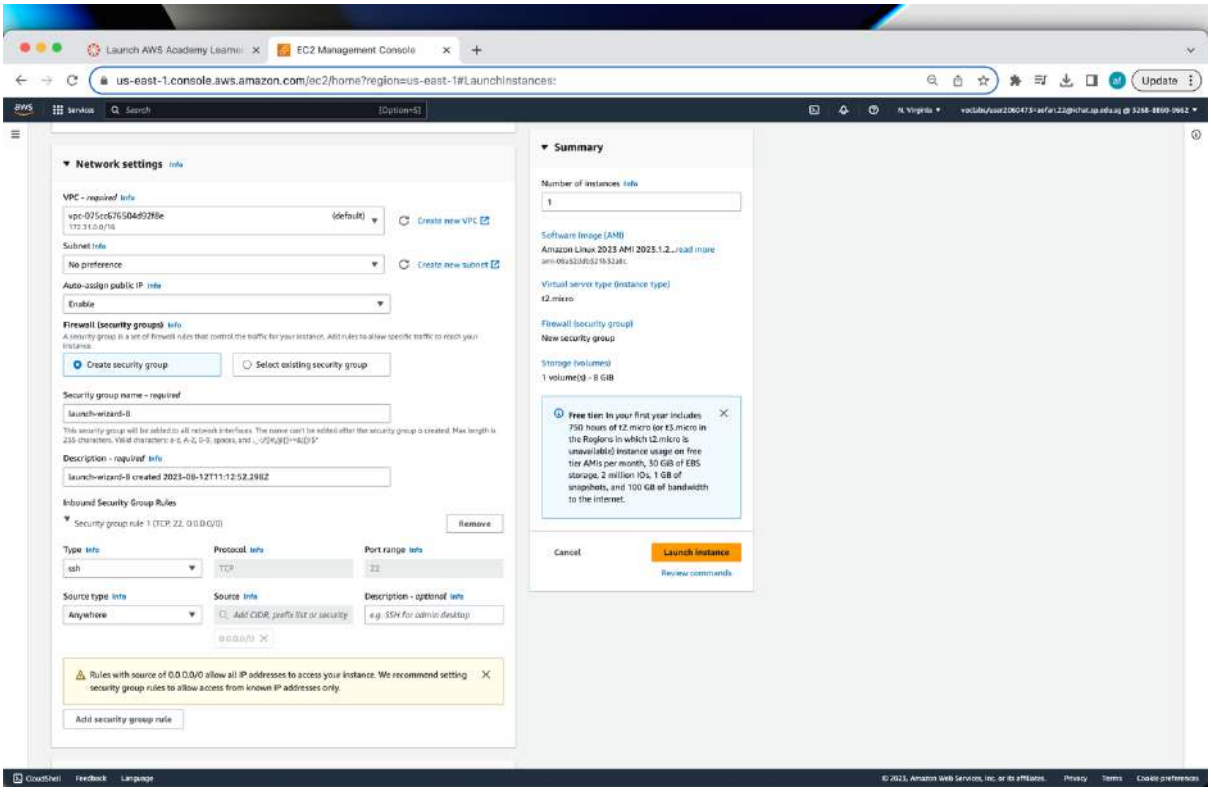
- For Instance type: Free tier eligible.



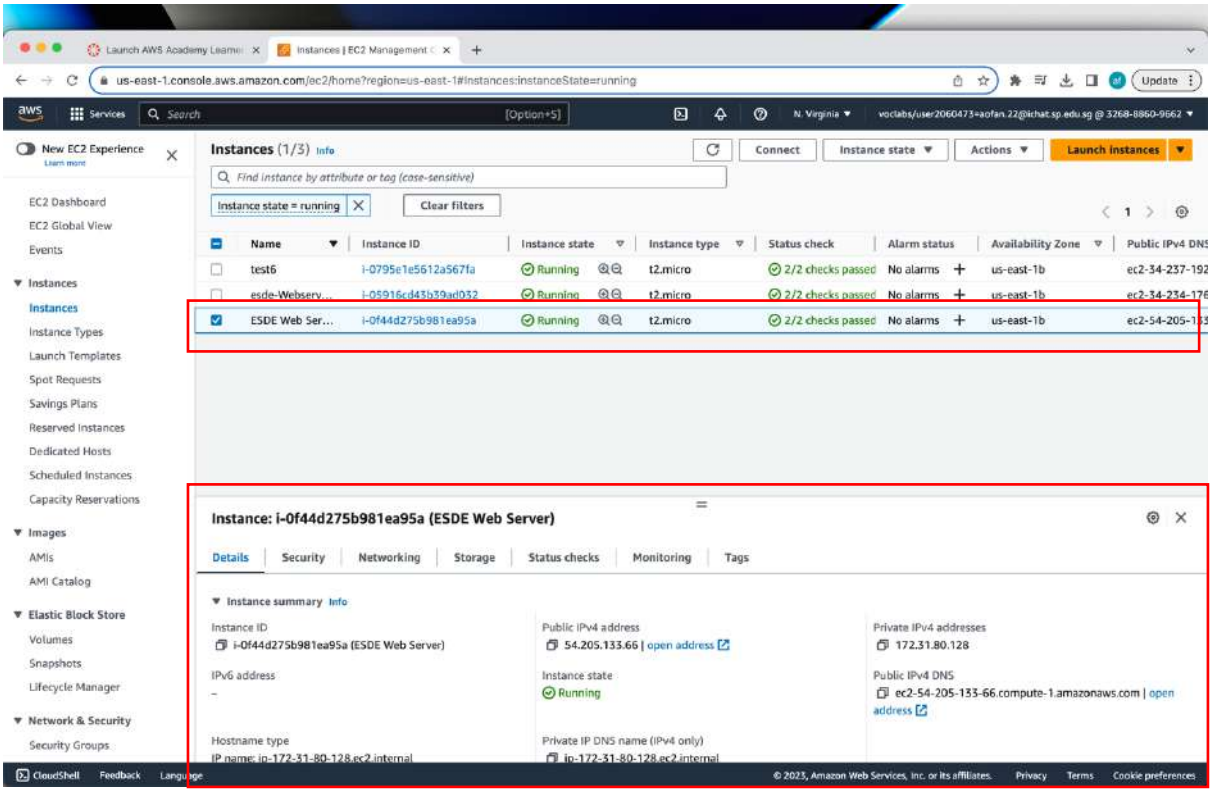
- For Key pair (login): Create new key pair.



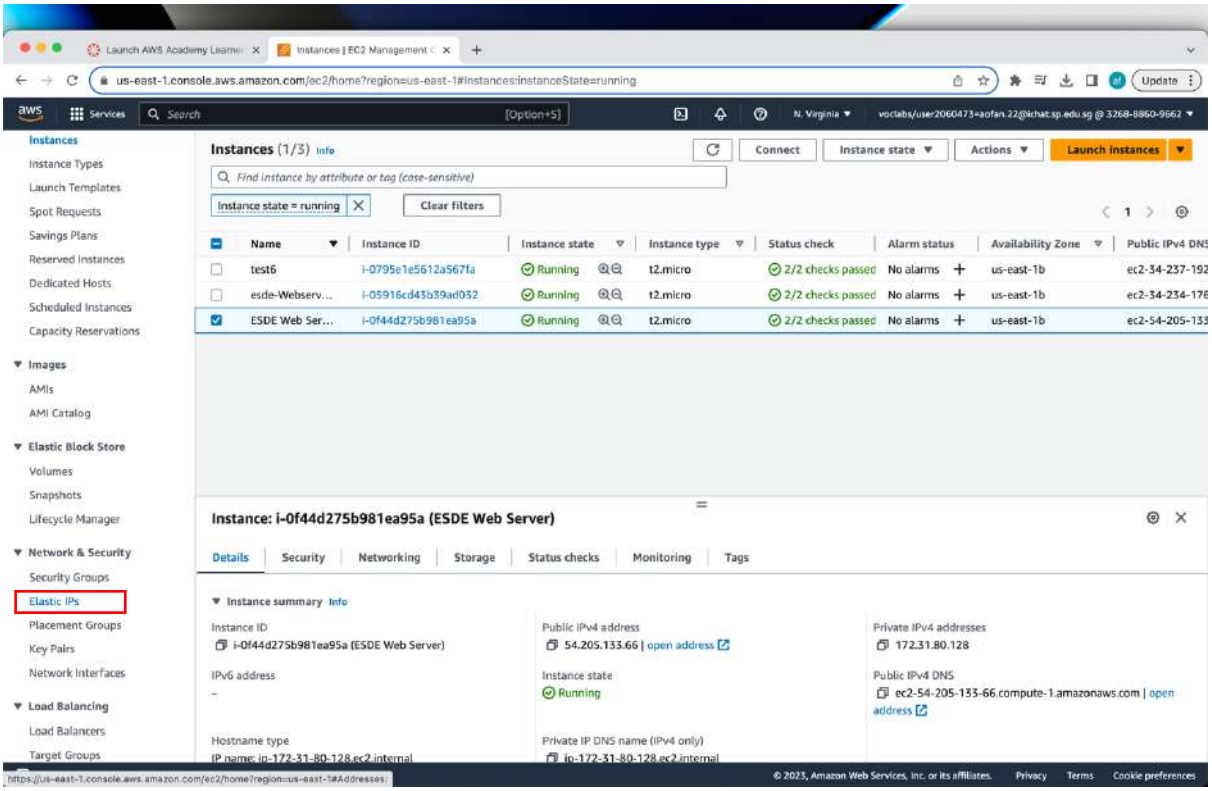
- For Network settings:



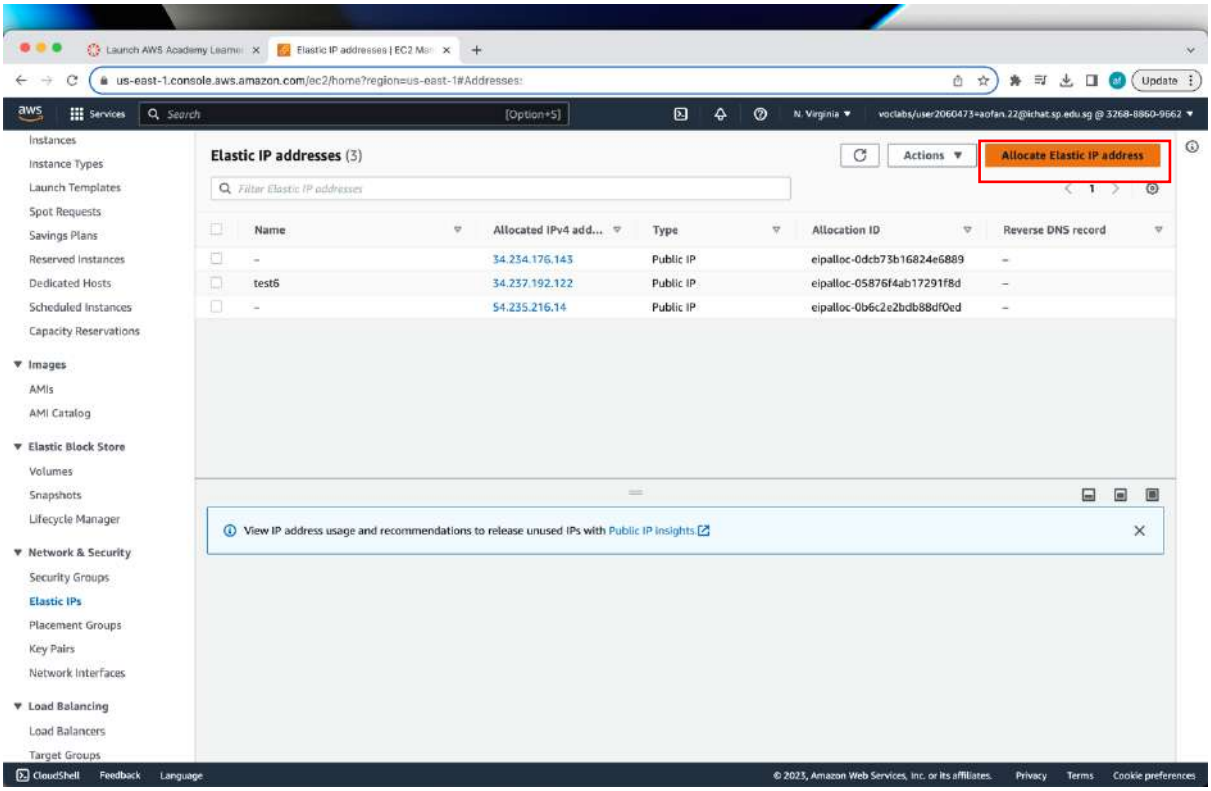
- Leaving everything else as default and click Launch instance. Navigate to Instances to confirm that the instance was created.



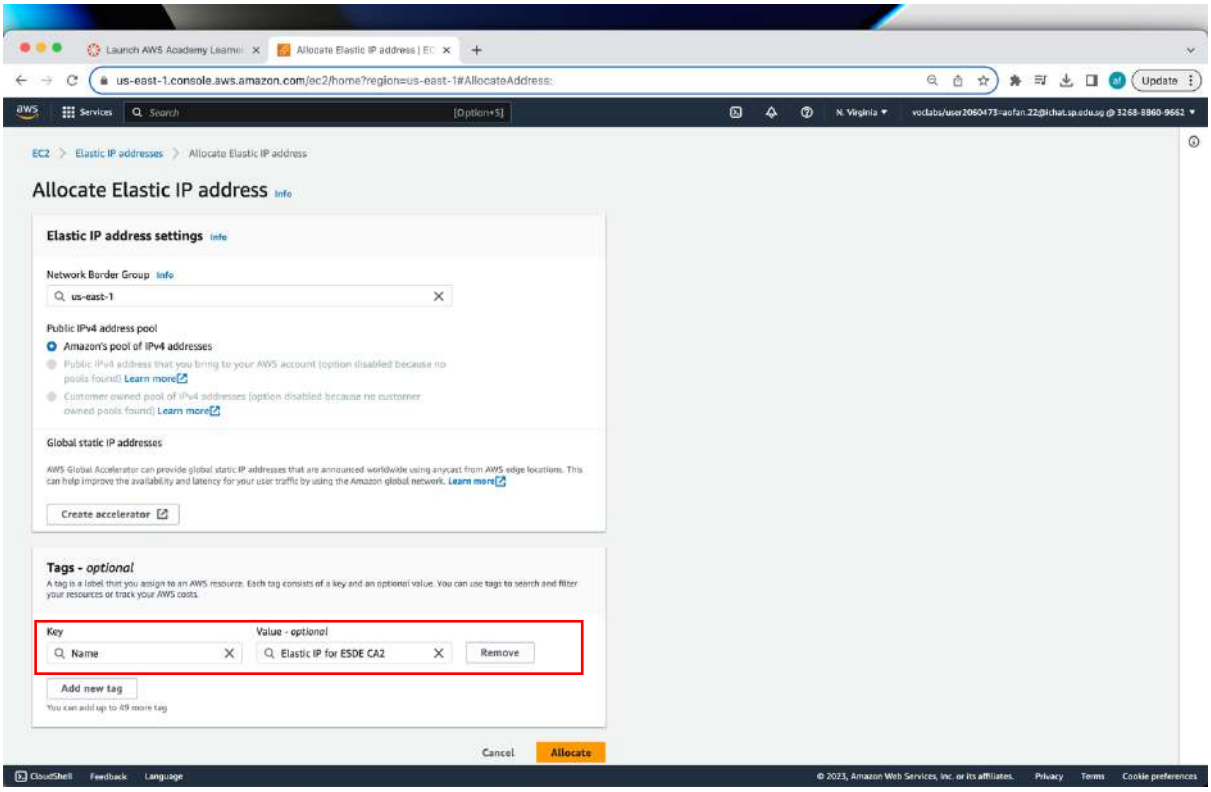
- After creating an EC2 instance, click on Elastic IPs under Network & Security from the left navigation bar.



- Click on Allocate Elastic IP address to create an elastic ip address.

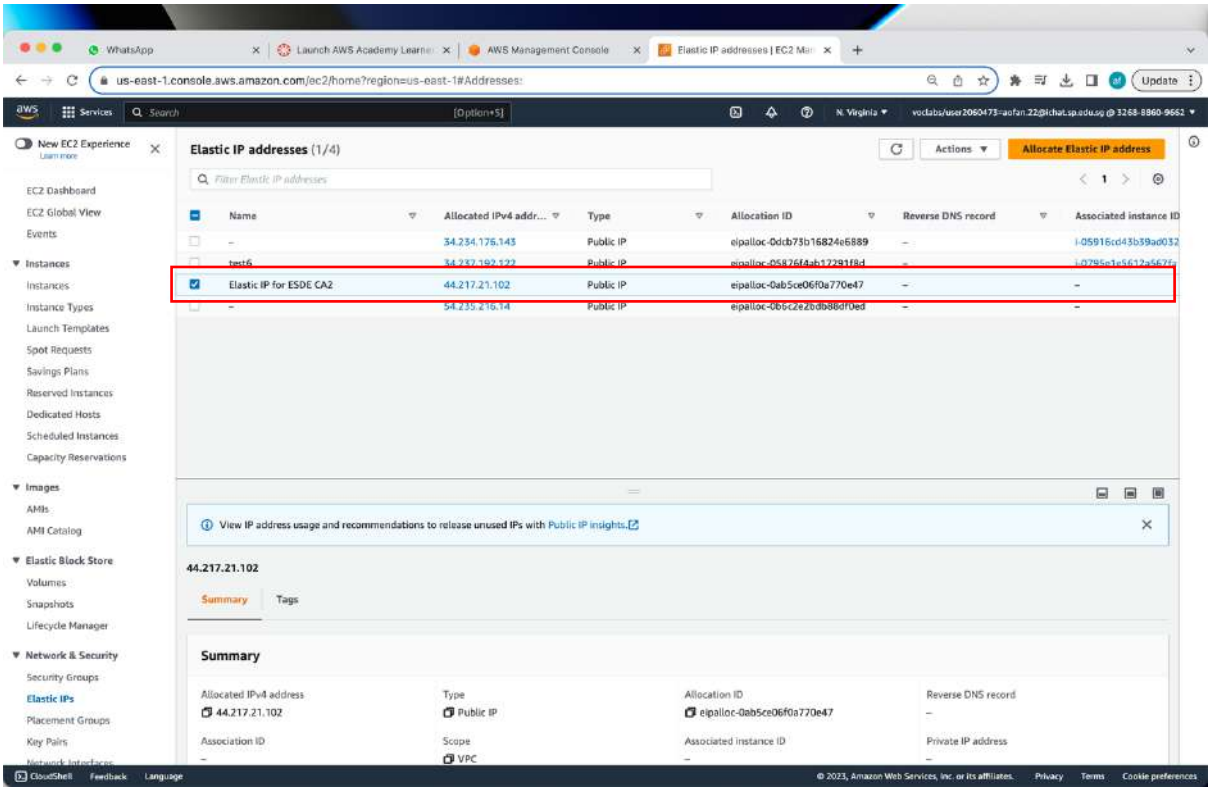


- For Tags:
  Key: Name, Value: Elastic IP For ESDE CA2.

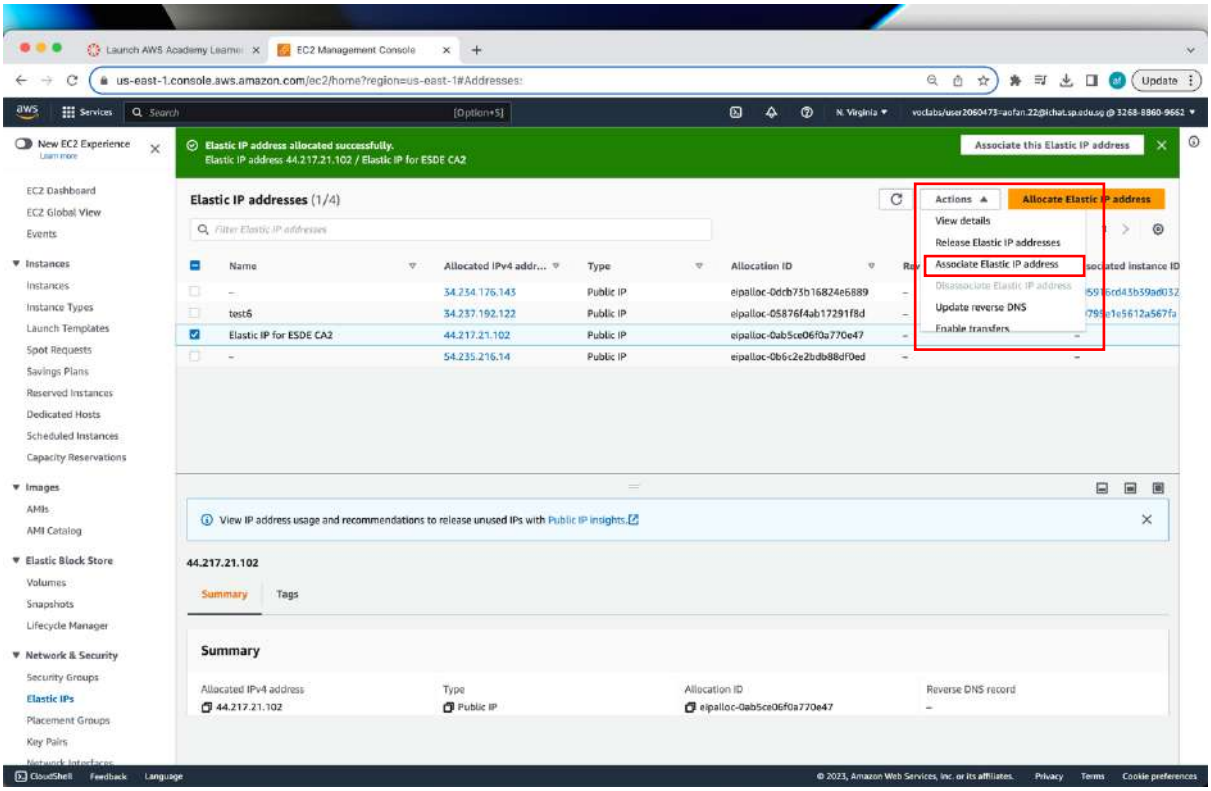


- Click on the Allocate button to finish creating the elastic ip.

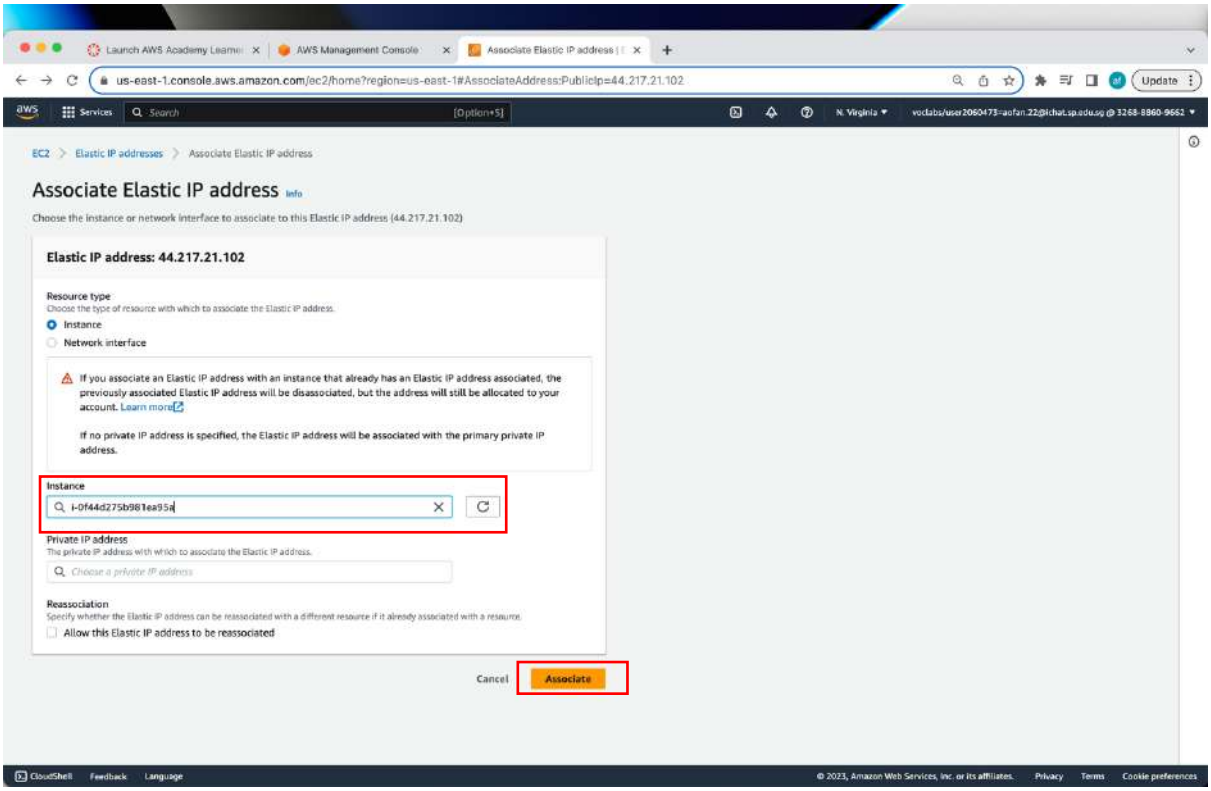
- Head back to Elastic IP addresses tab to confirm that the creation of the elastic ip was successful.



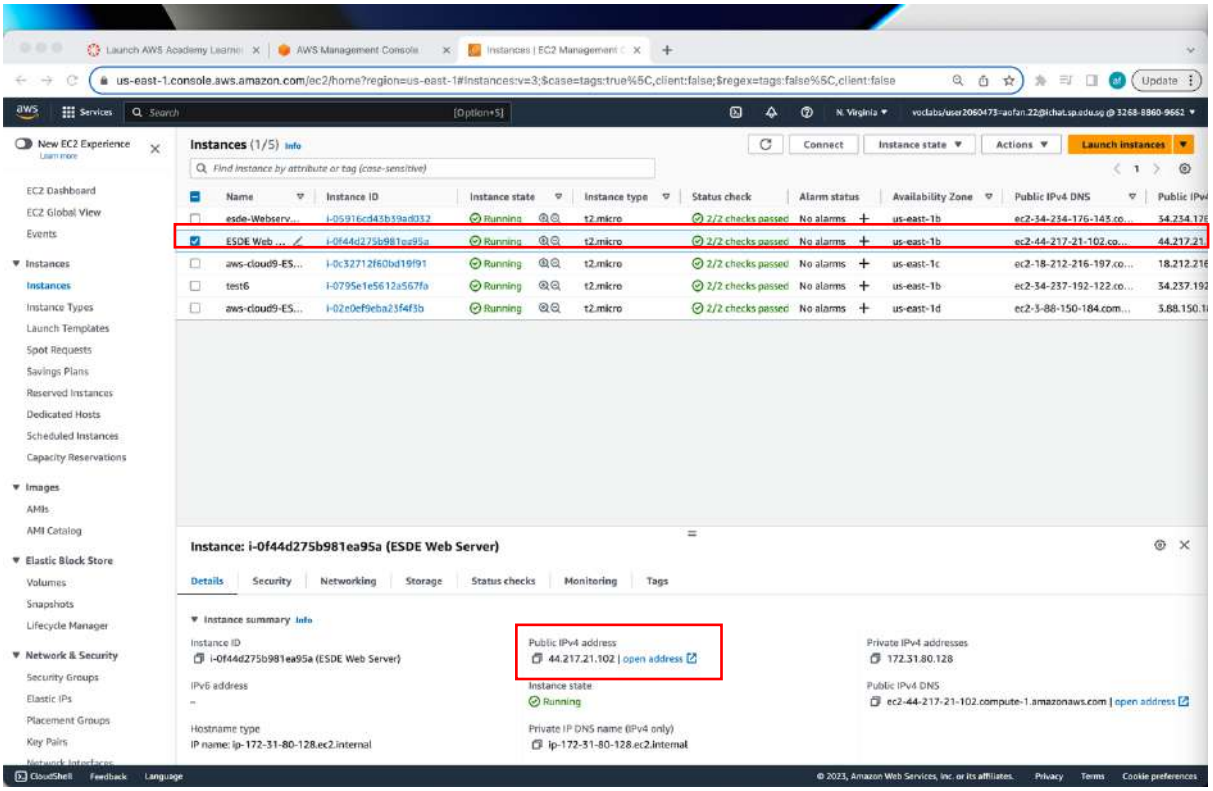
- Click on the Actions drop down menu and select Associate Elastic IP address.



- Choose the instance created earlier and click the Associate button.

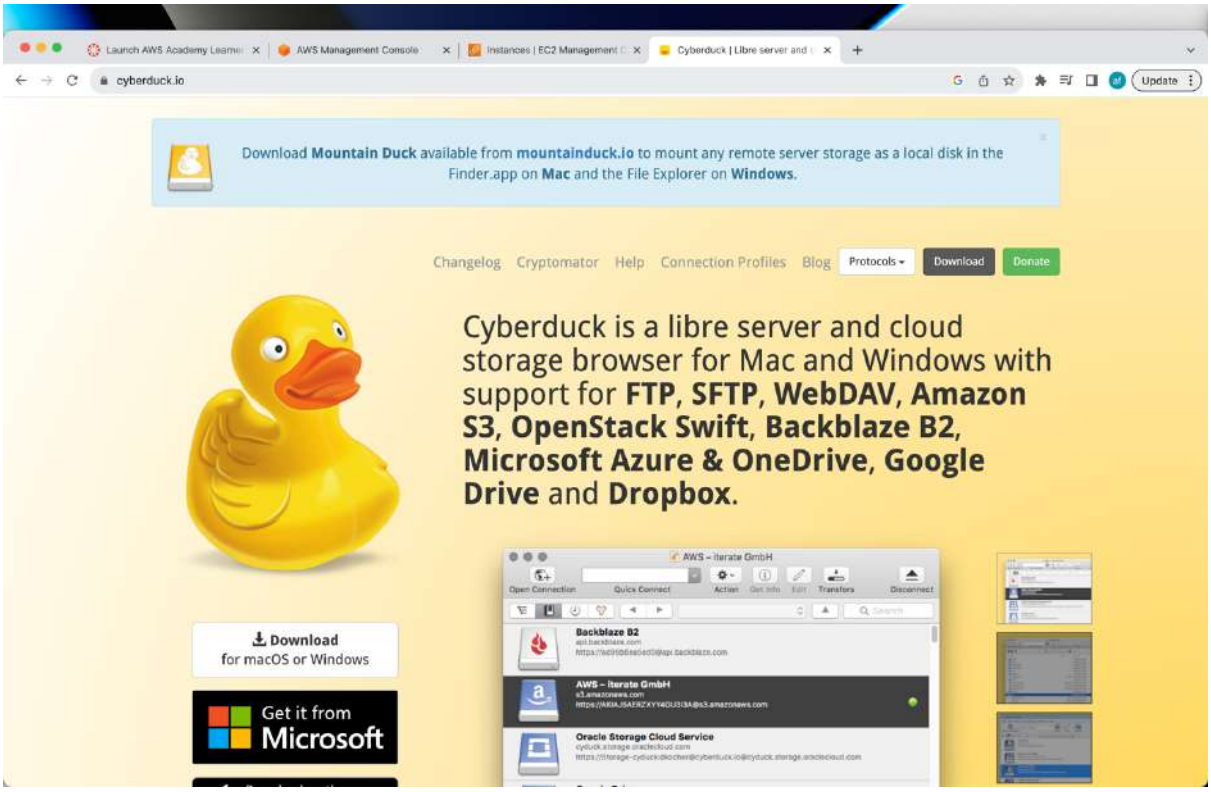


- Head back to the Instances tab to check the public IPv4 address of the instance that was just associated with the Elastic IP. In this case, it's 44.217.21.102.

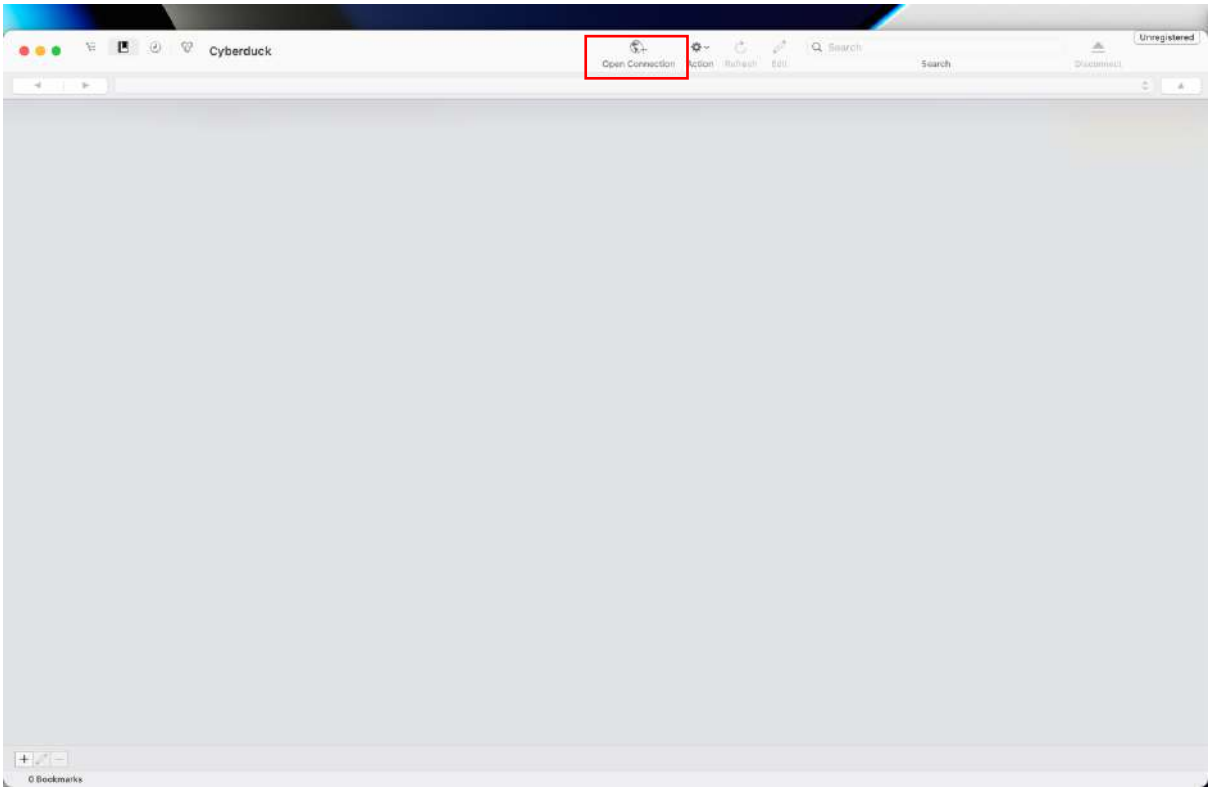


- Copy the public IPv4 address as it is needed for connecting to the instance.

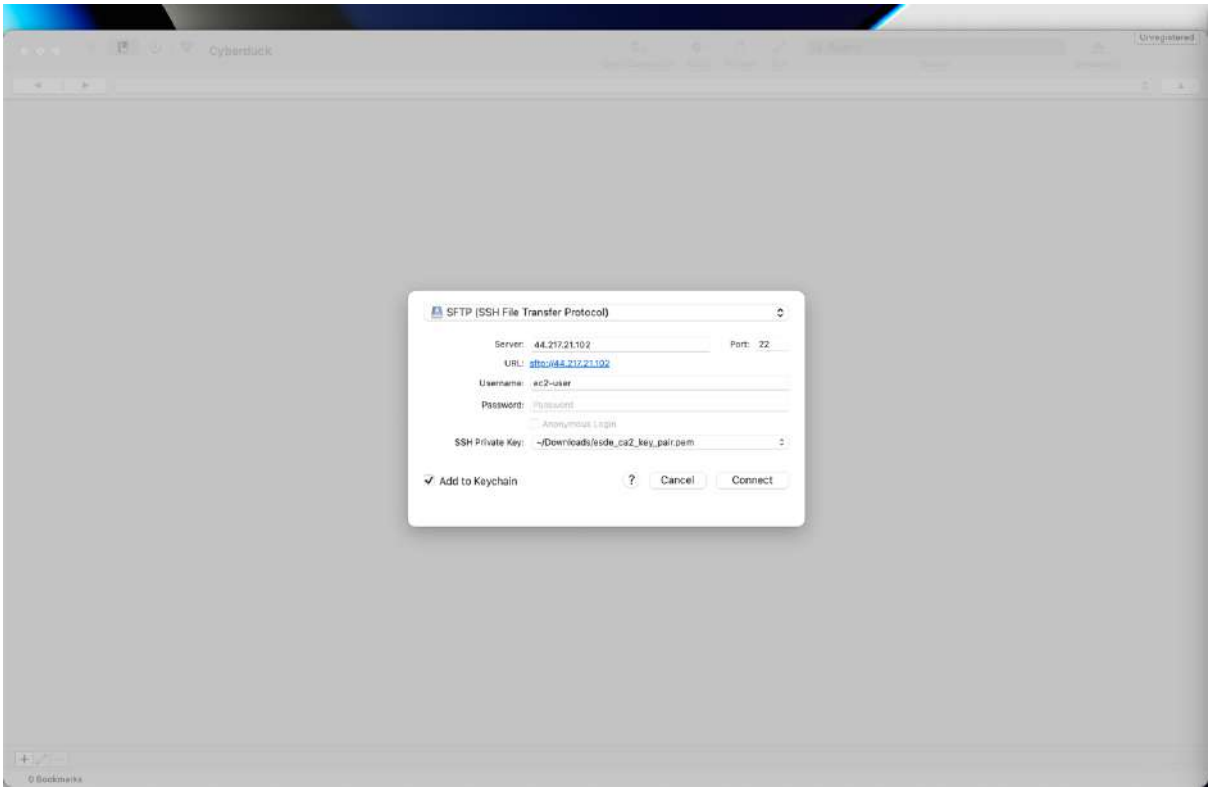
- Download Cyberduck and open it.



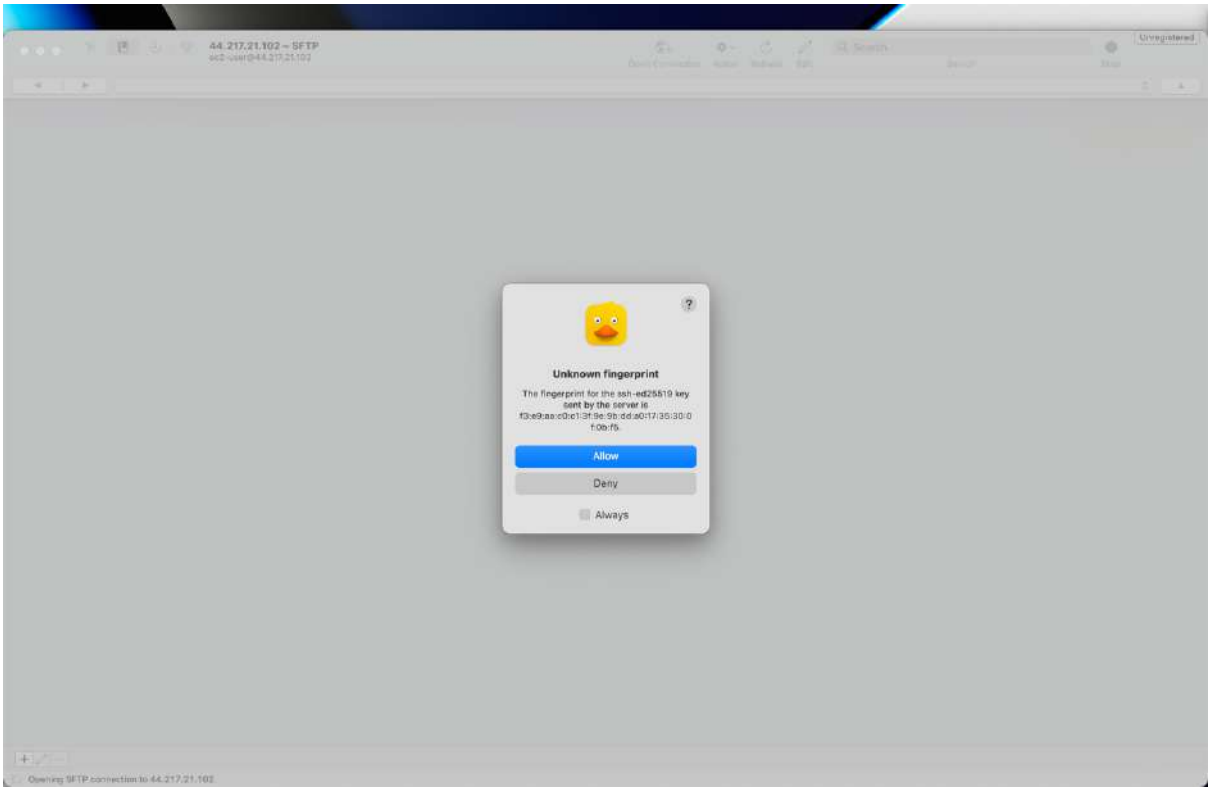
- Click on Open Connection to connect to the instance.



- Select SFTP (SSH File Transfer Protocol), paste the public IPv4 address into Server:, type ec2-user in Username: and select the key pair that was downloaded during the creation of the instance in SSH Private Key:. Click Connect to proceed to the next step.

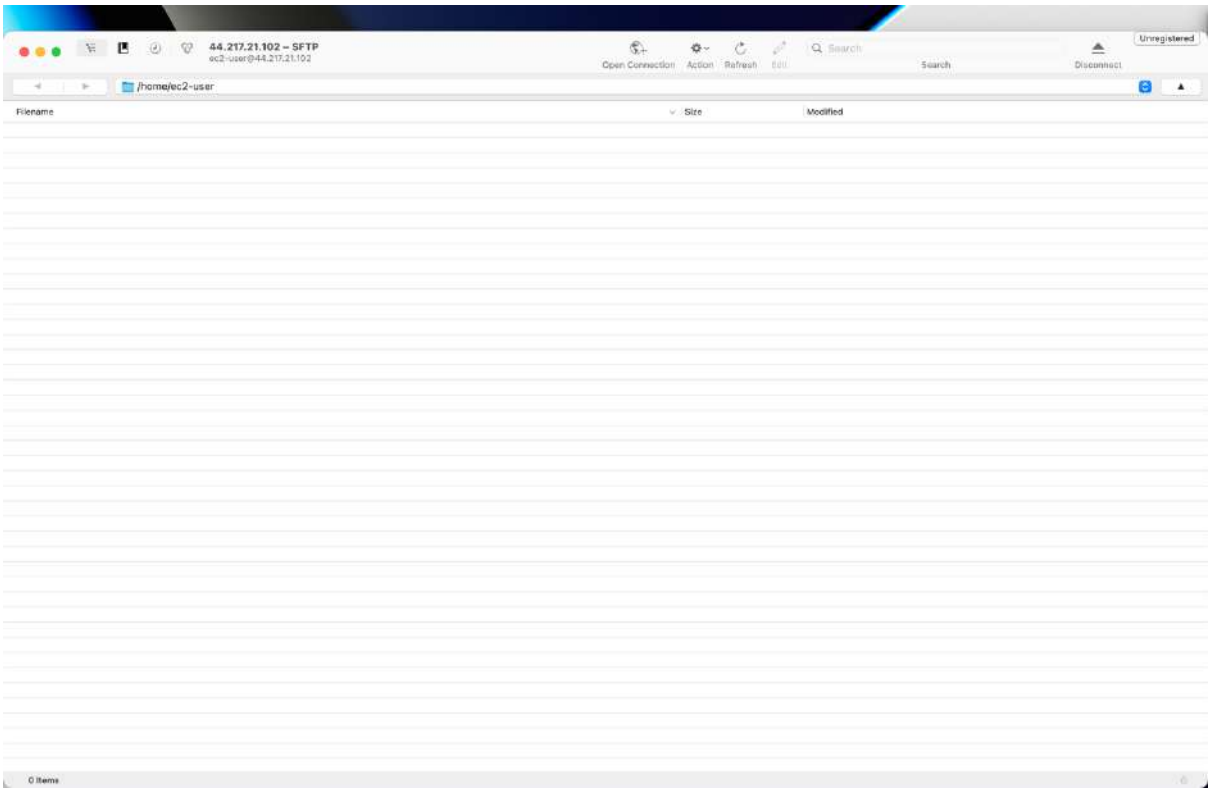


- Choose Allow when Cyberduck prompt for Unknown fingerprint.

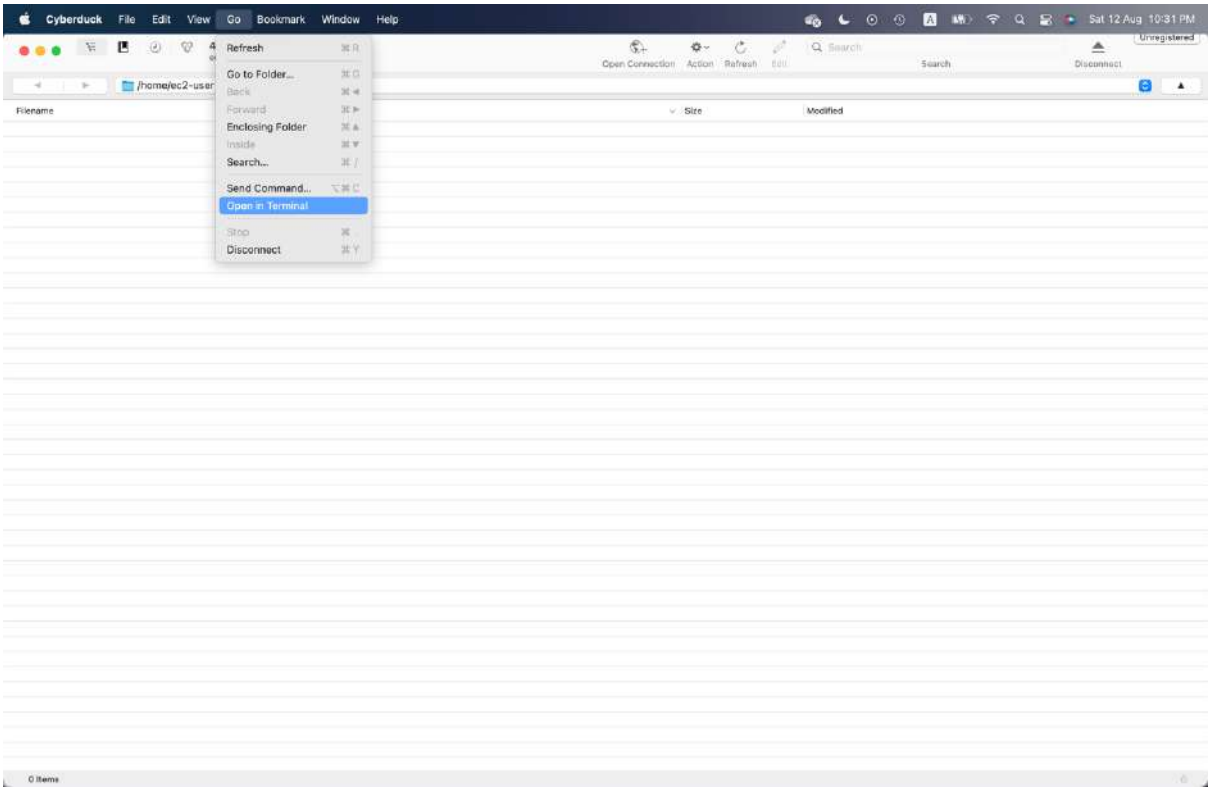

44.217.21.102 – SFTP
Unregistered
Unknown fingerprint
The fingerprint for the ssh-ed25519 key sent by the server is
Allow
Deny
Always
Opening SFTP connection to 44.217.21.102

3. Installing NodeJS on EC2 Instance

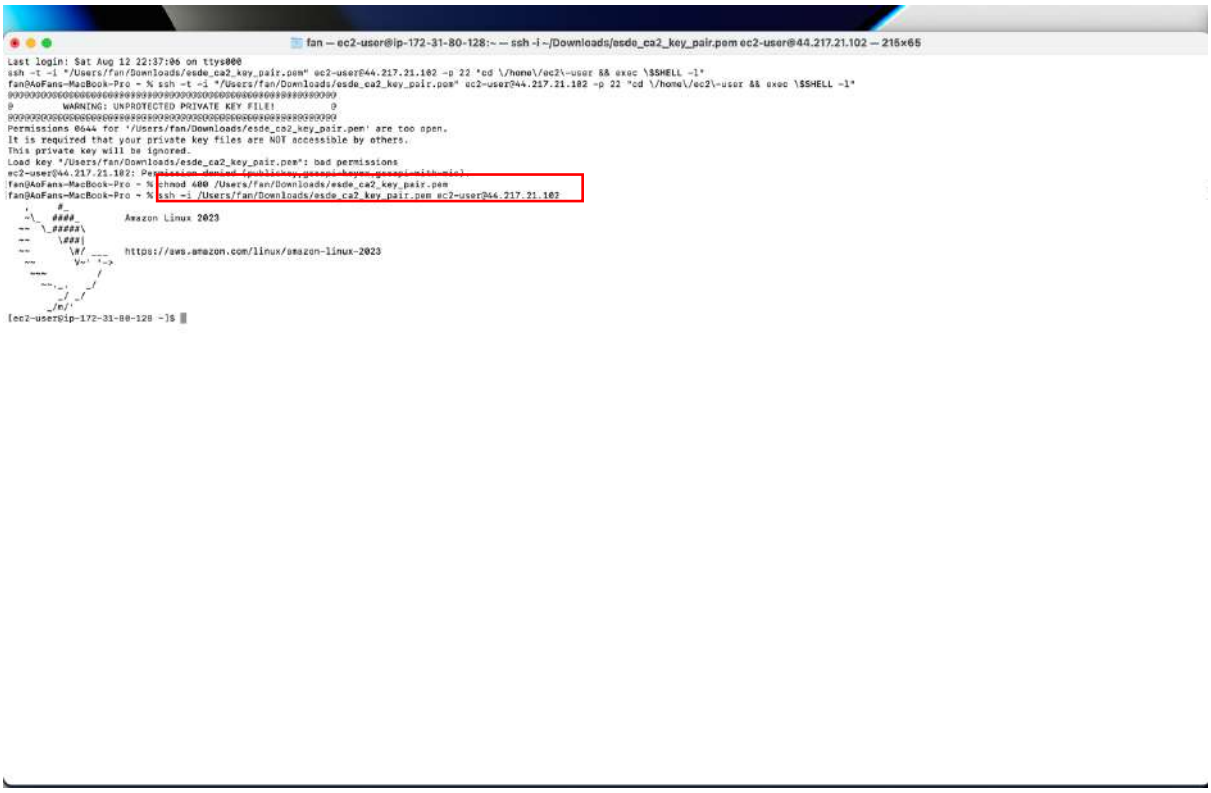
- Upon successfully connecting to the instance, you will be presented with a page similar to the illustration below.



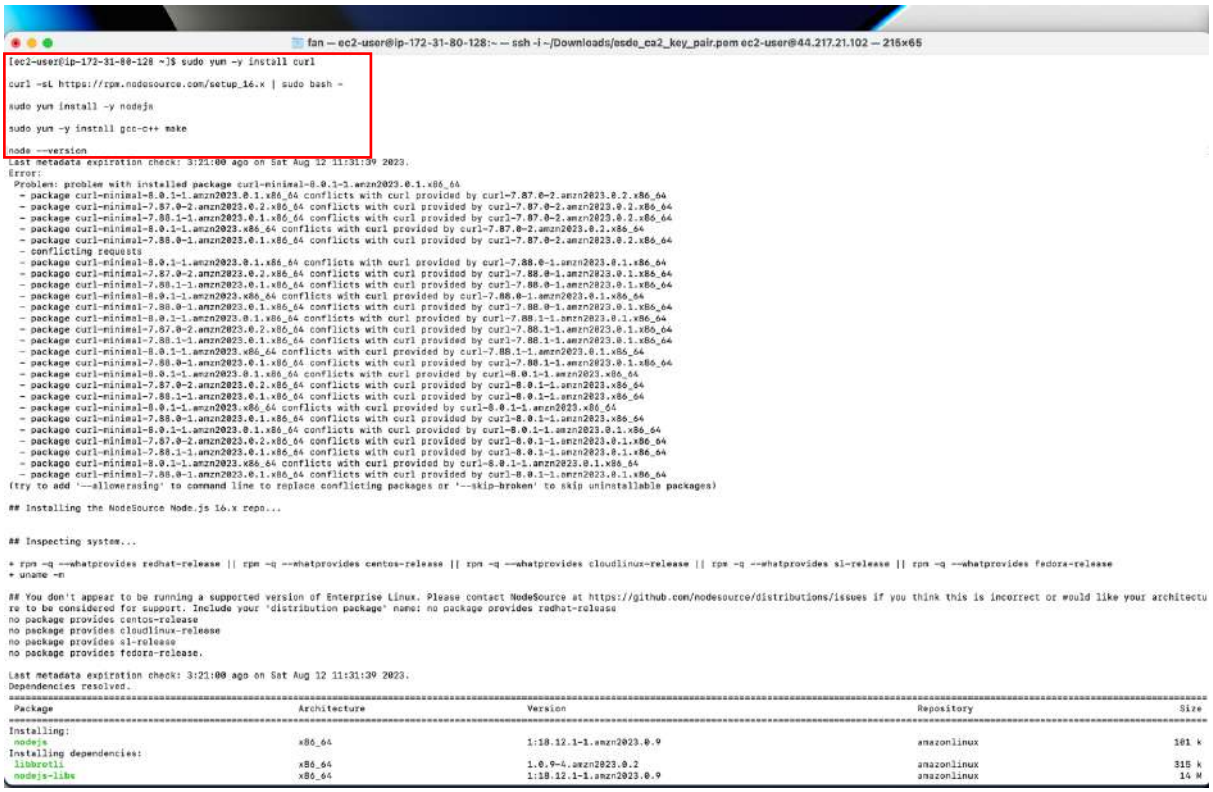
- Open the terminal in Cyberduck via the menu bar.



- Enter chmod 400 + the key pair (drag the pem file into the terminal). Followed by ssh -I + the key pair + ec2-user@ + the public IPv4 address.

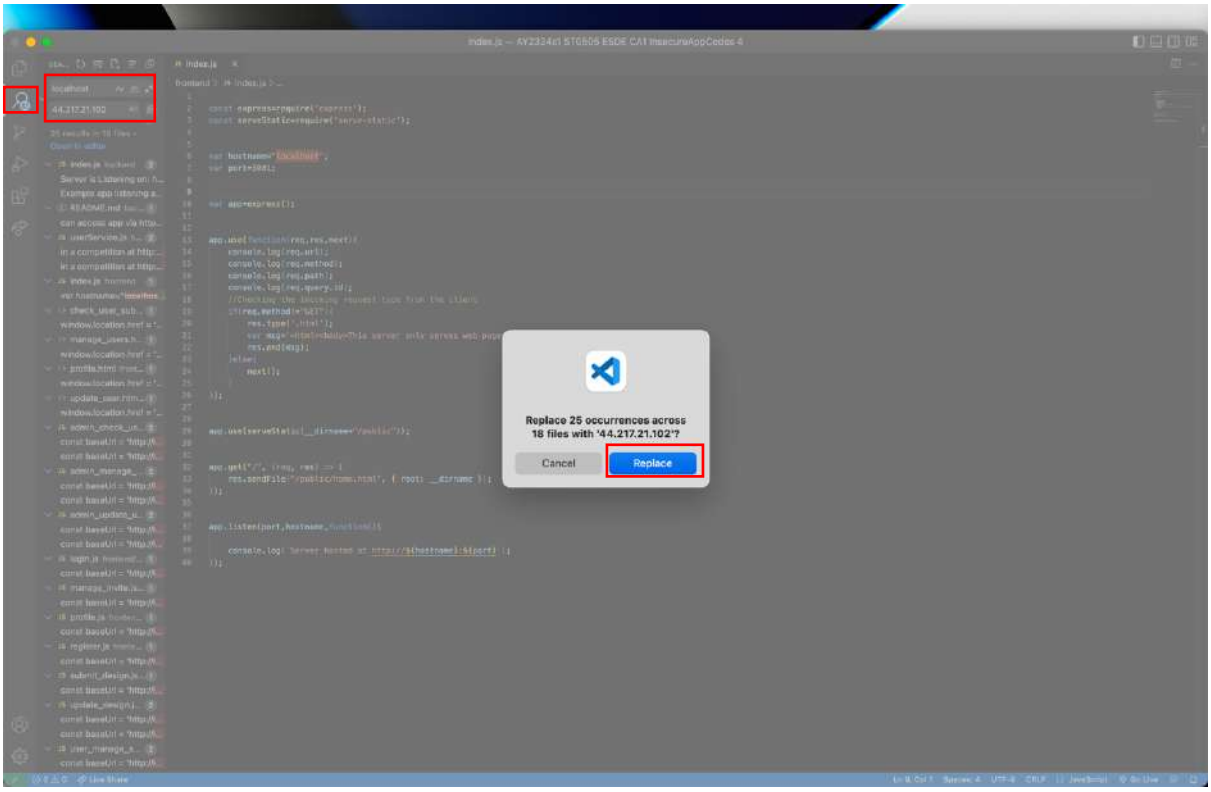


- Run
  sudo yum -y install curl
  curl -sL https://rpm.nodesource.com/setup_16.x | sudo bash -
  sudo yum install -y nodejs
  sudo yum -y install gcc-c++ make node –version.

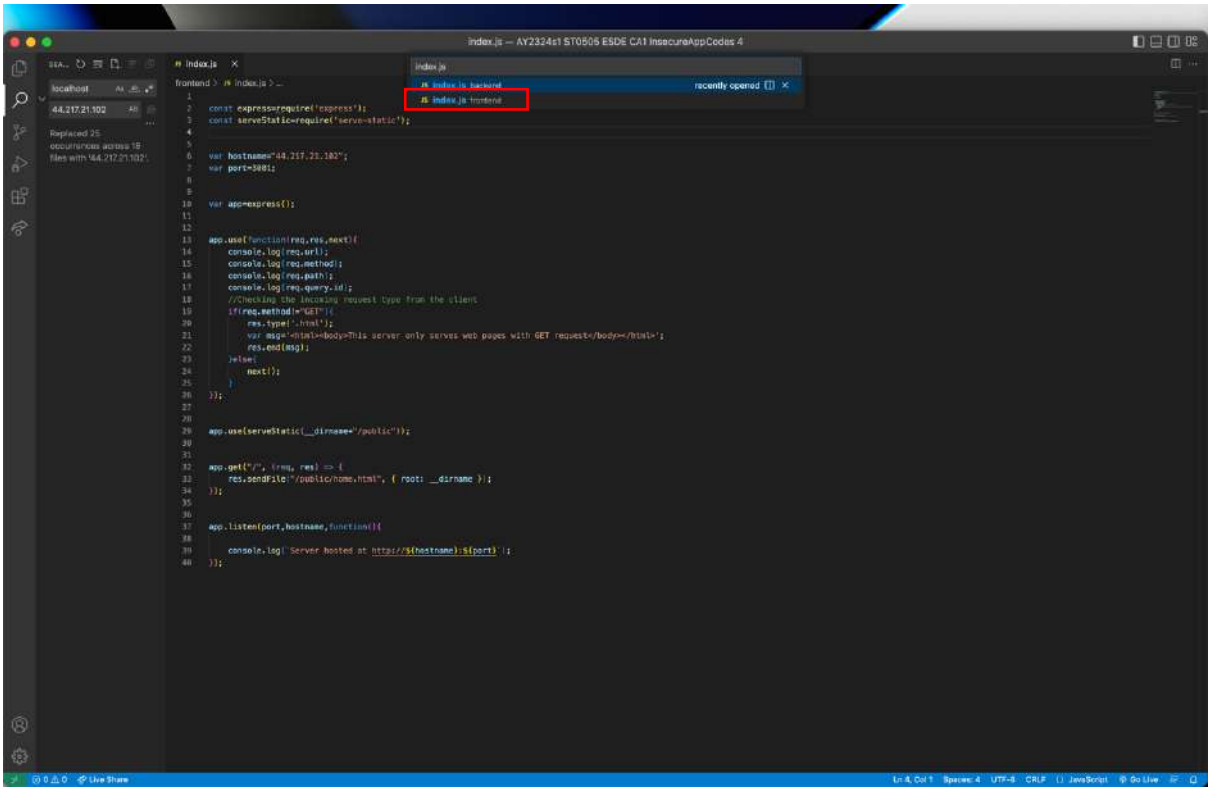


4. Deploying ‘BeeDesign’ app on EC2 Instance

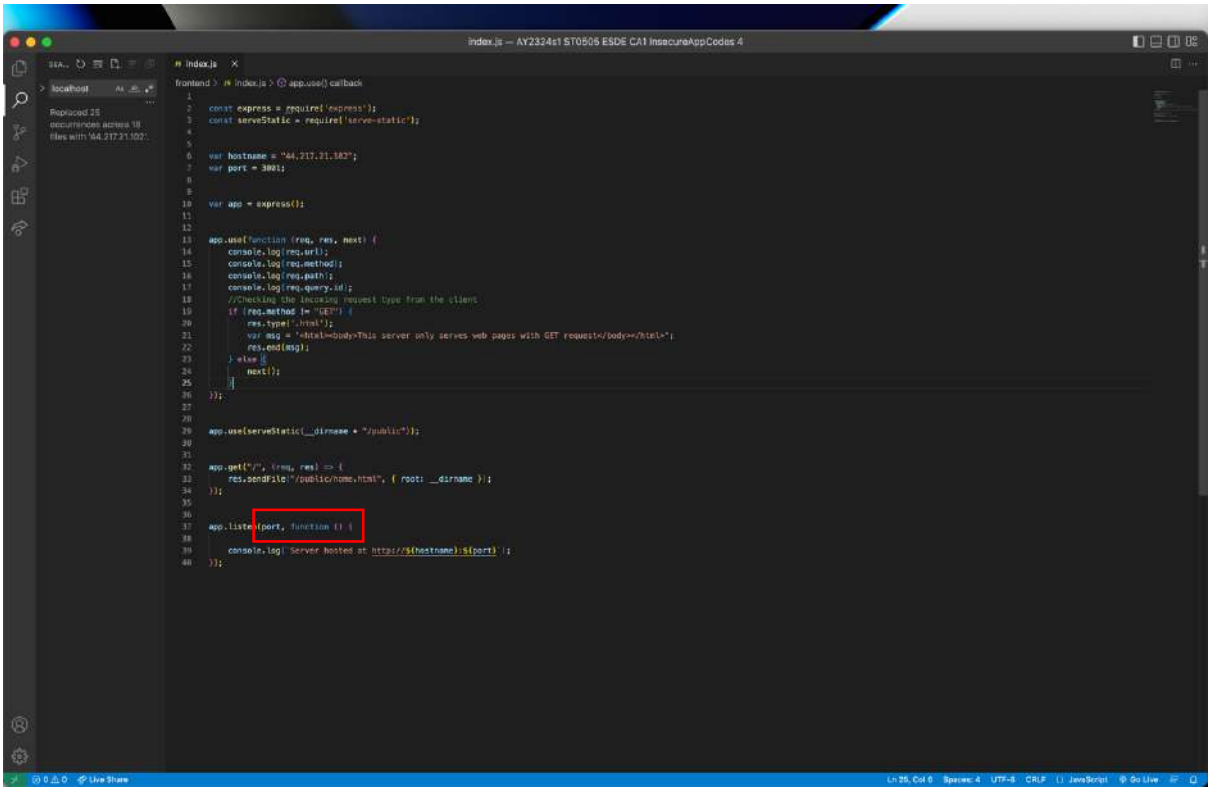
- Open the original bee design app files with Visual Studio Code. Click on the magnifying glass icon on the left or Command + Shift + F to search for localhost. Replace localhost with elastic ip address (public IPv4 address).



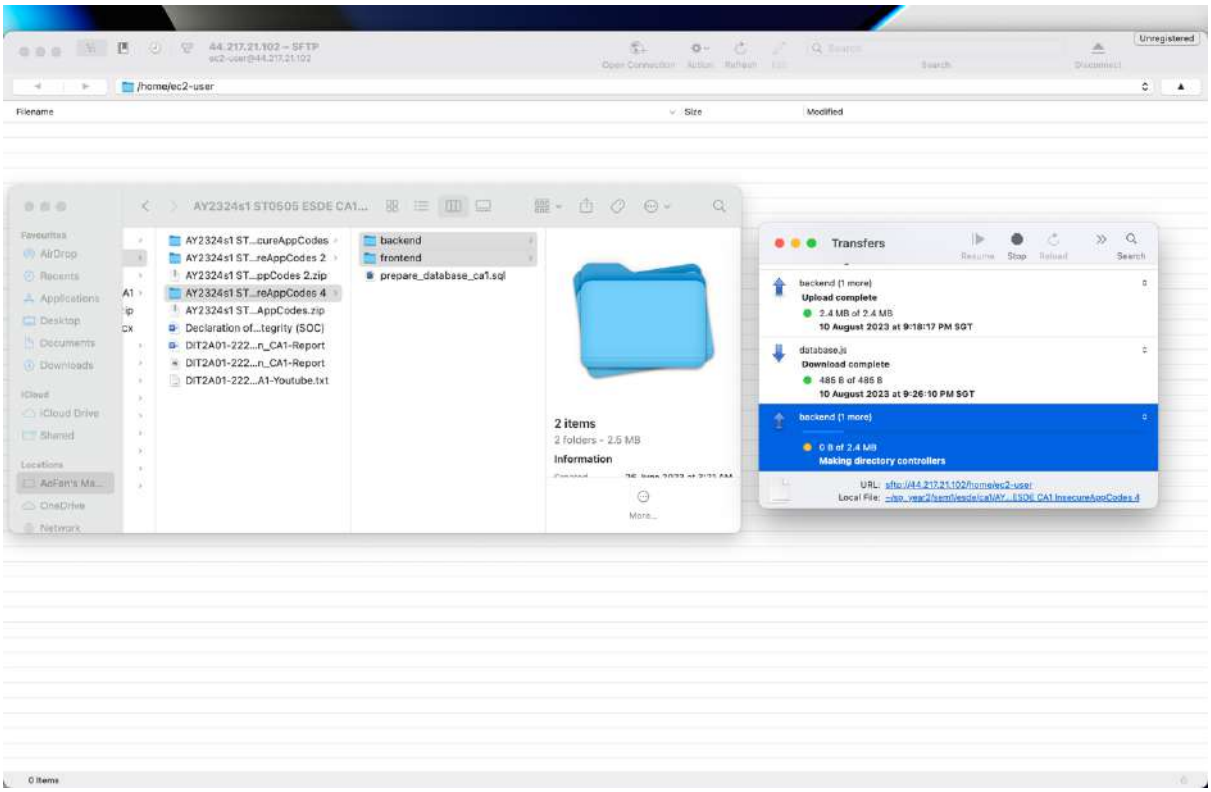
- Command + P to search for index.js file in the front end.

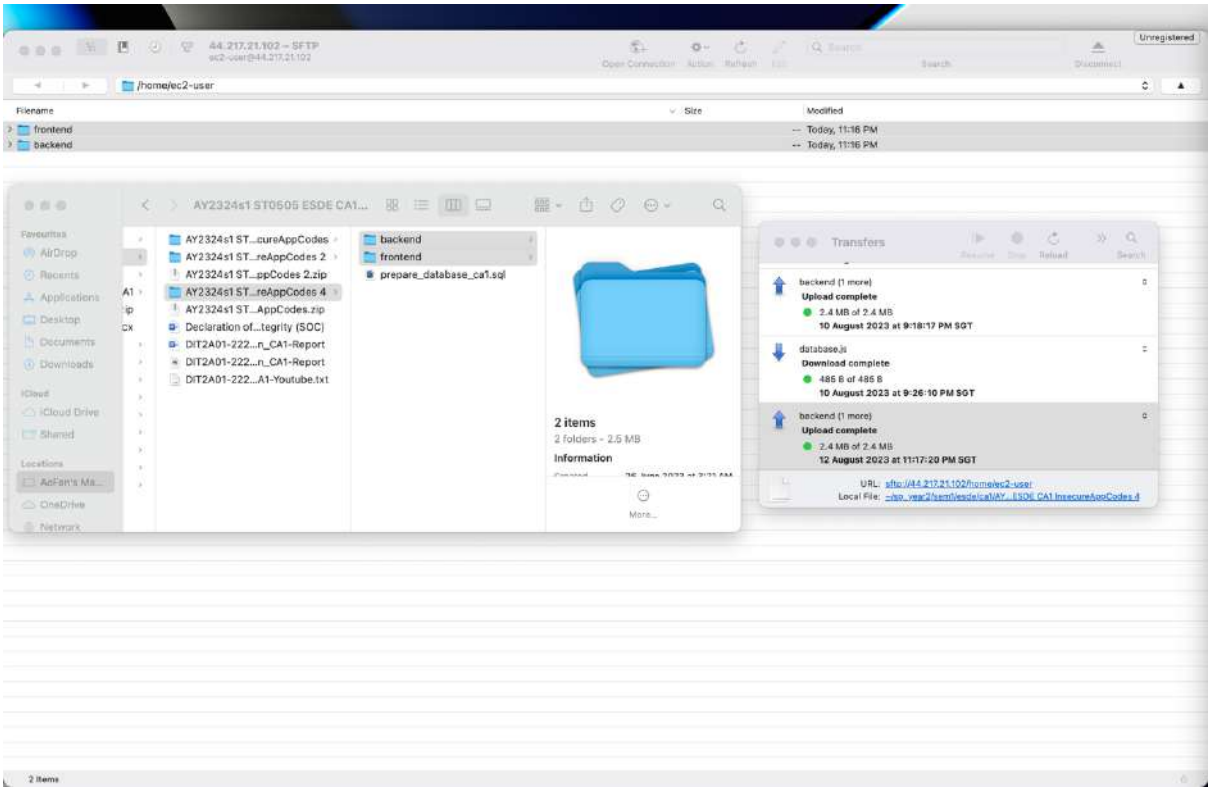


- Remove hostname from the app.listen parameter. (line 37)

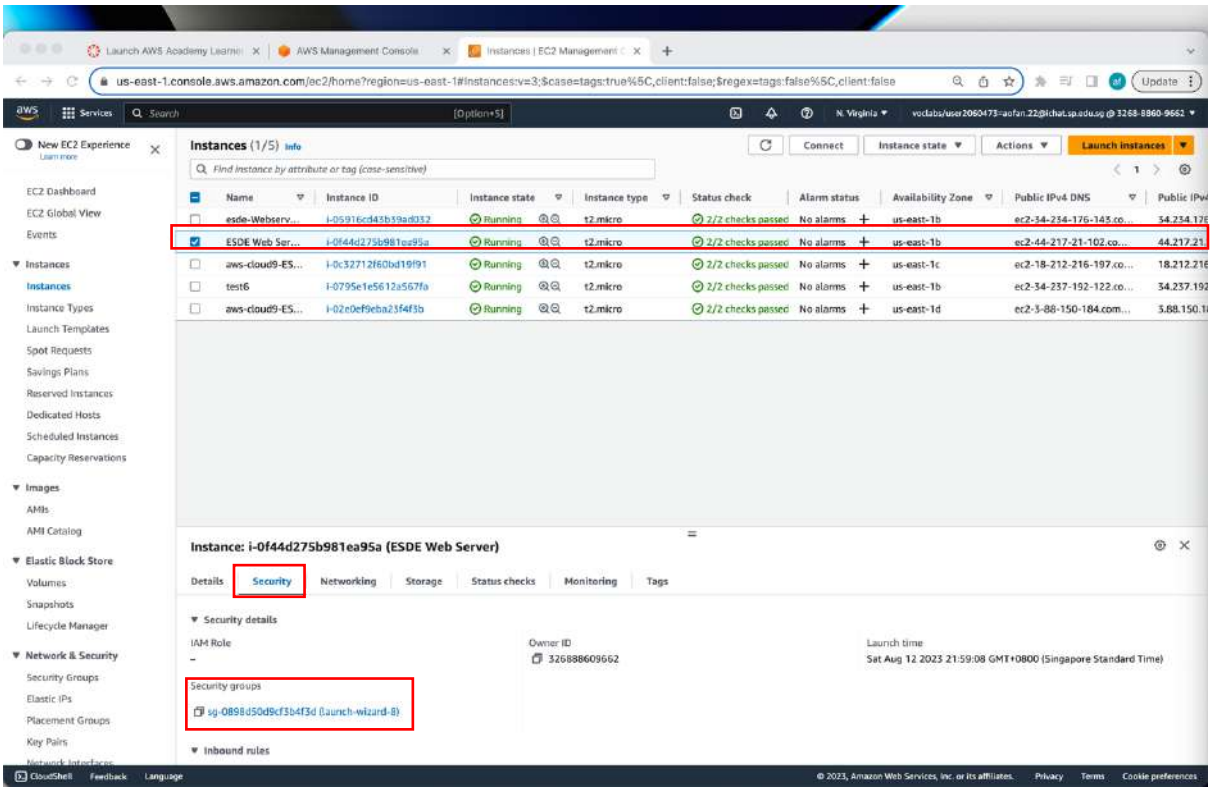


- Upload the local bee design app files on your local machine to the instance. (Copy the files in the finder and paste them in Cyberduck)

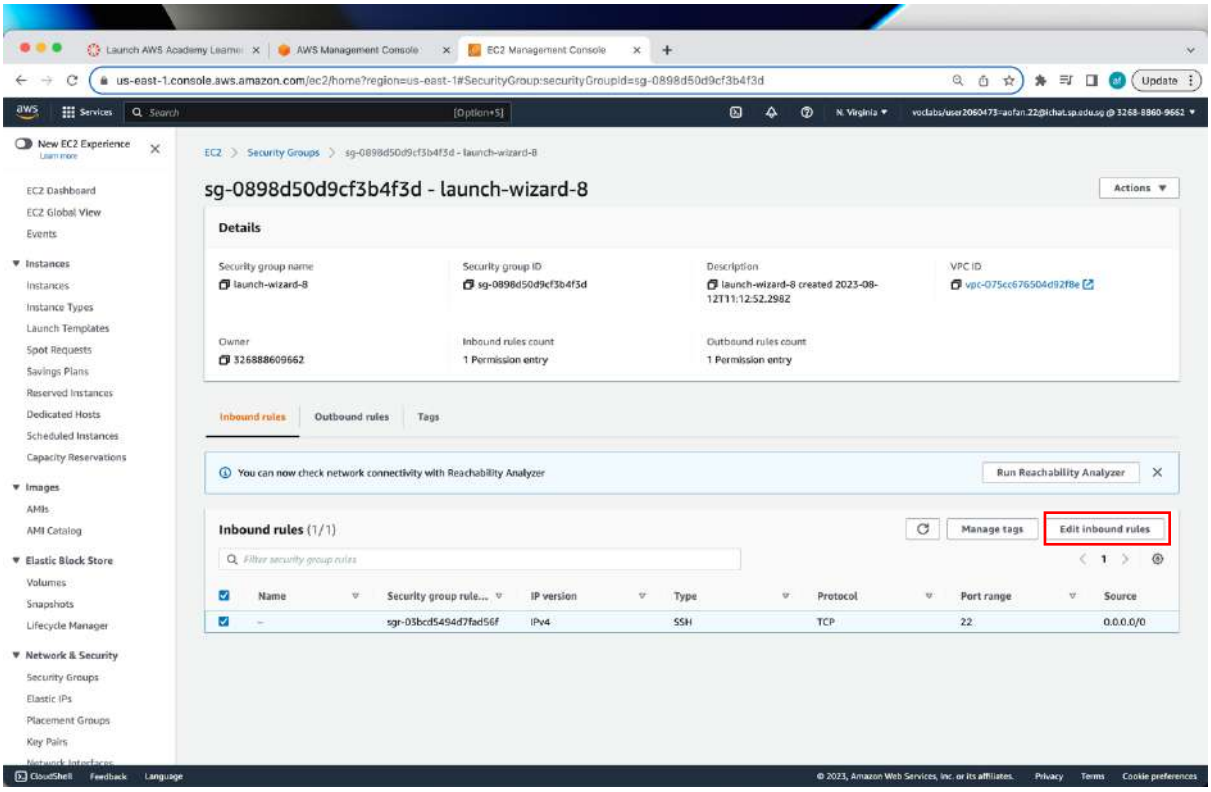




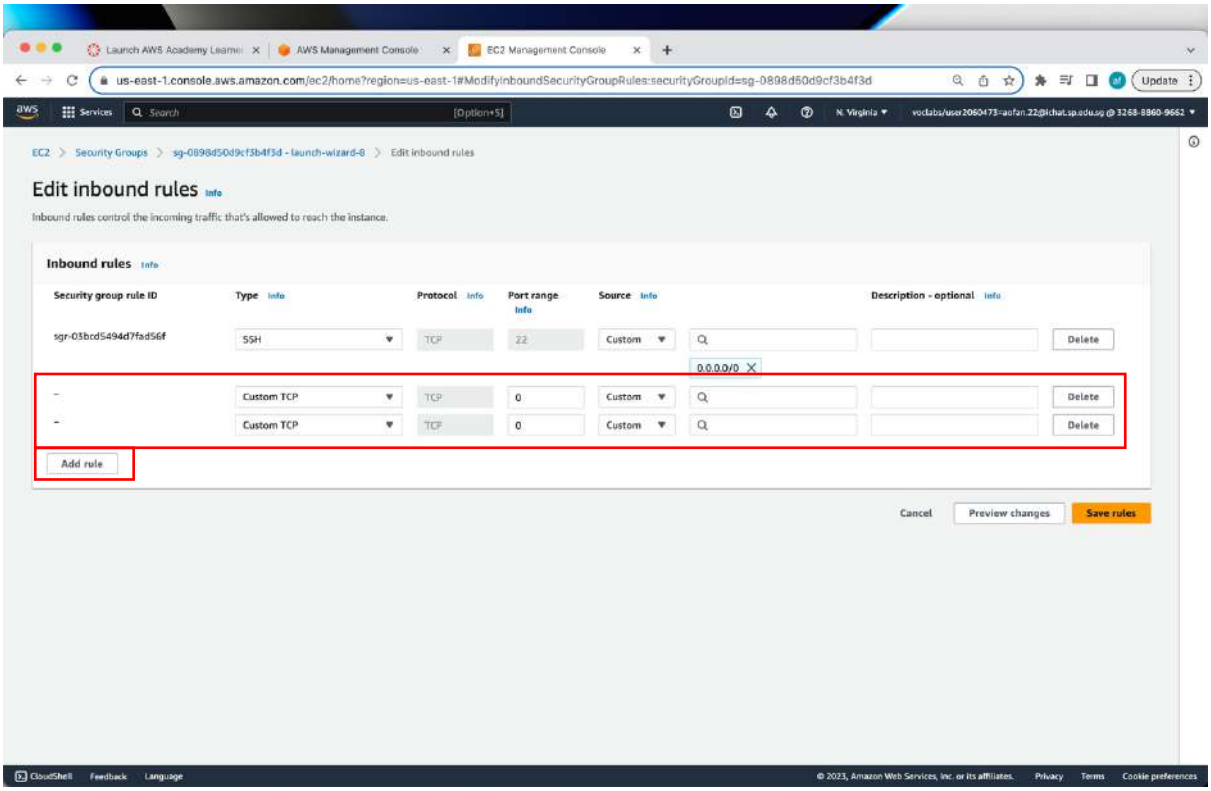
- Navigate to AWS, select ESDE Web Server instance and click on Security, followed by Security groups.



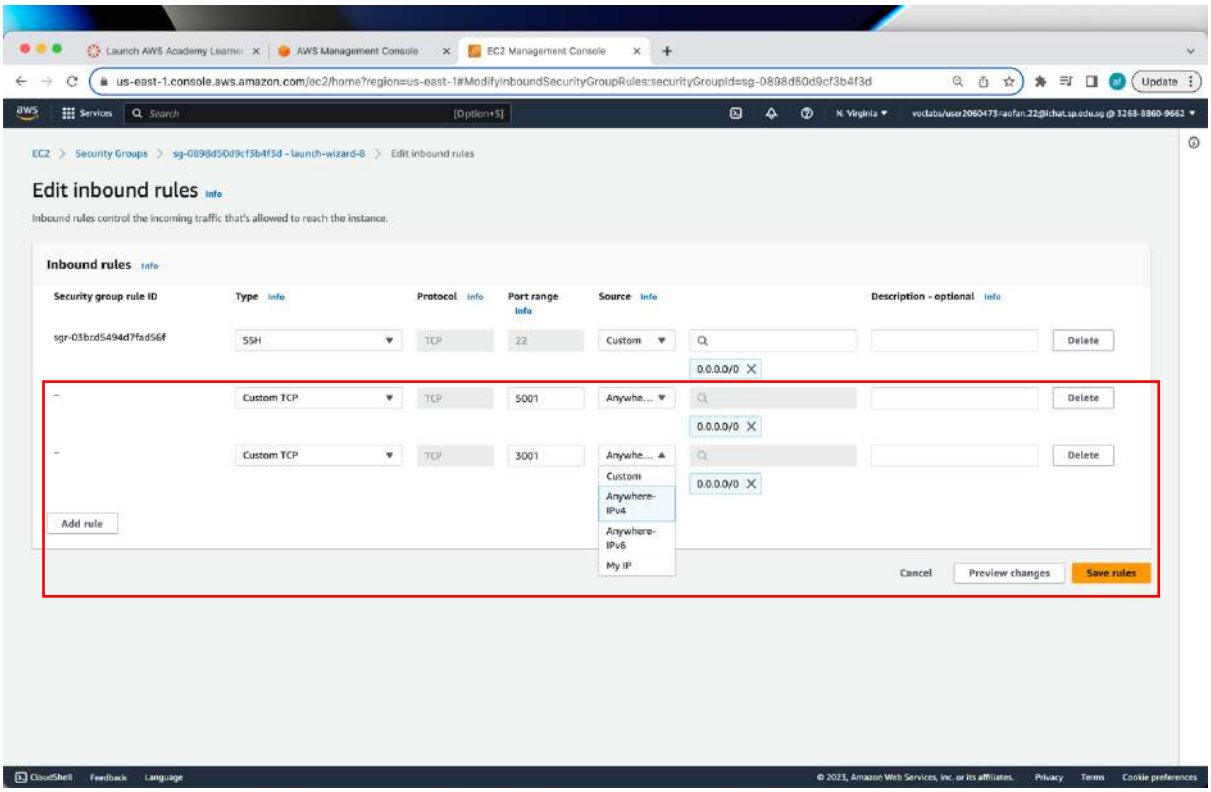
- Click on the Edit inbound rules button.



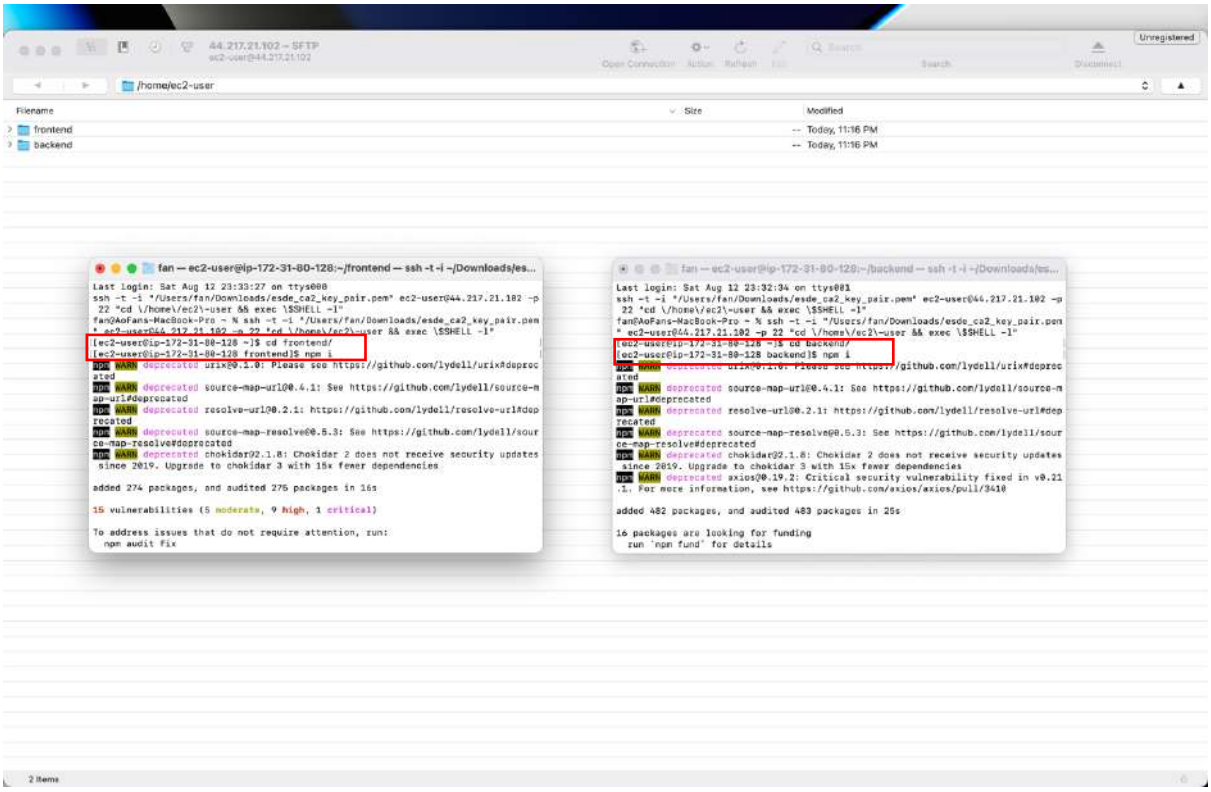
- Click Add rule to add rules for the back-end port (5001) and the front-end port (3001).



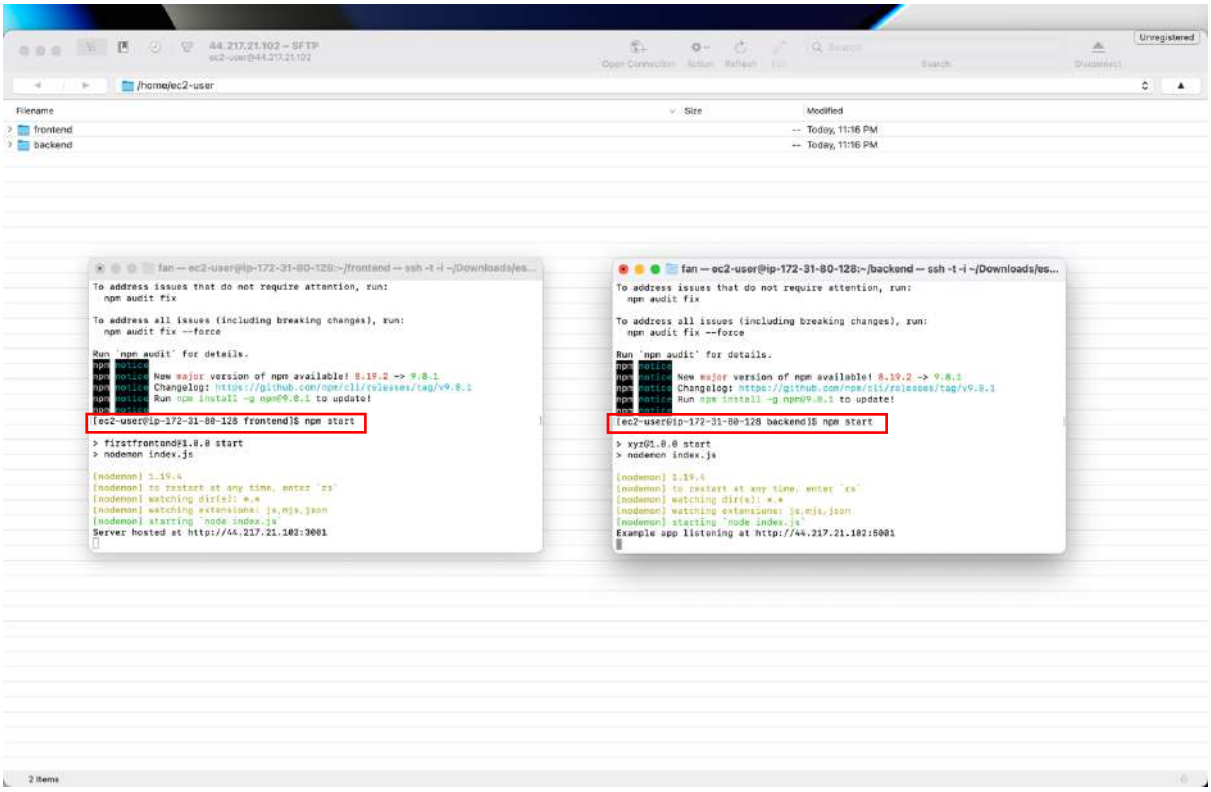
- Enter the respective port number under Port range and select Anywhere IPv4 under Source for both rules. Click Save rules to finalise the changes made.



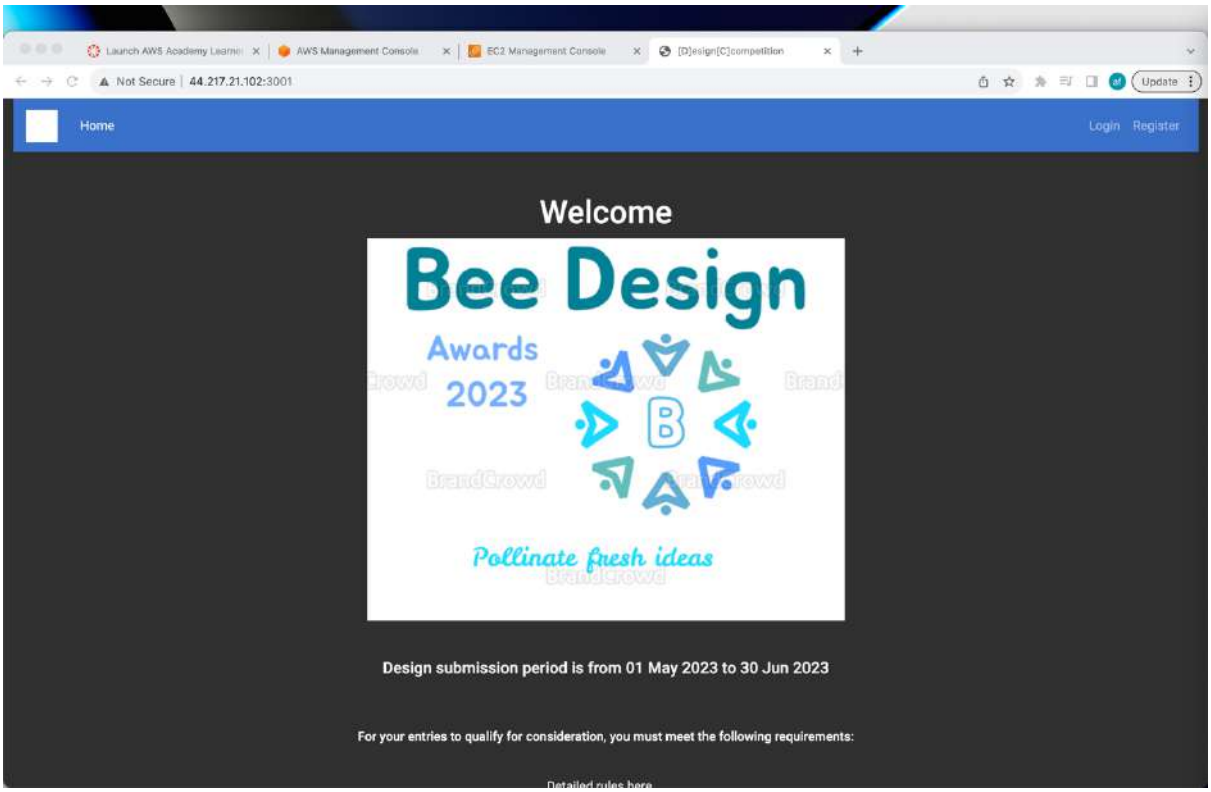
- Open 2 terminal in Cyberduck, one for front-end and one for back-end. cd to respect files and run npm i for both terminals.



- Run npm start for both terminals

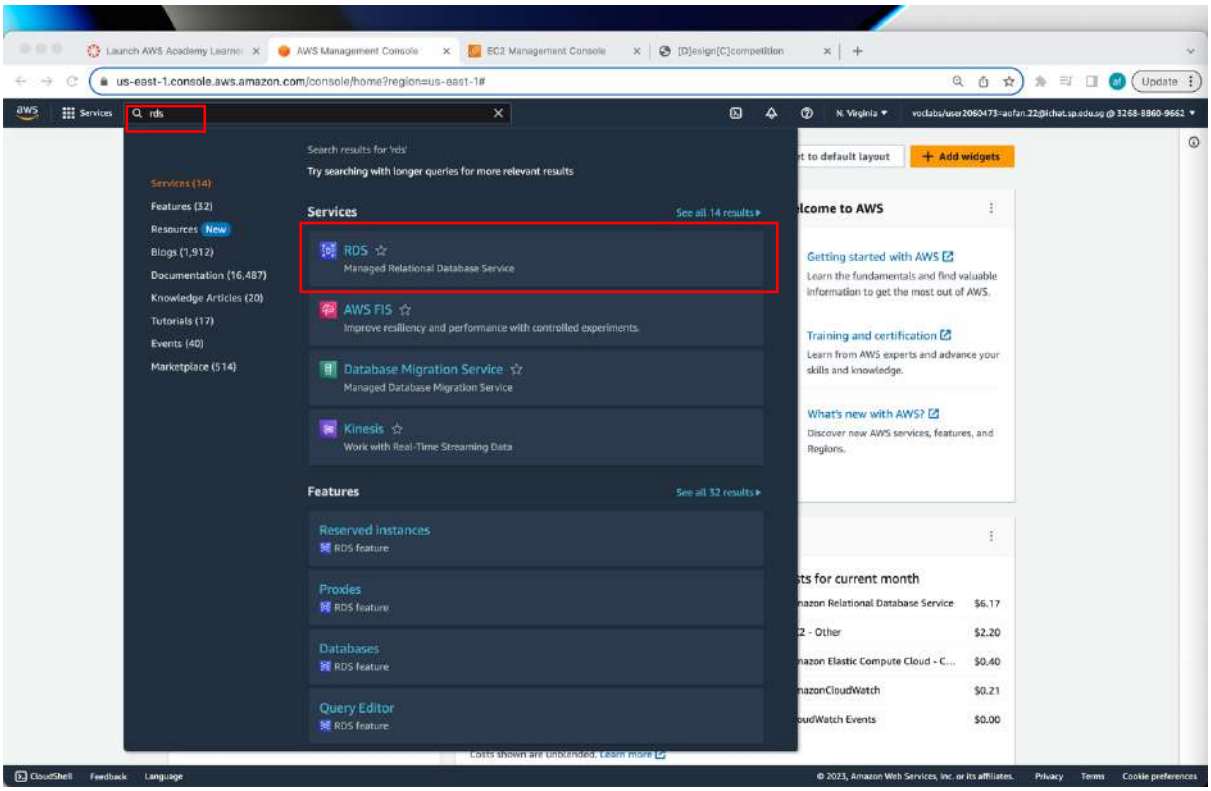


- Copy the URL displayed in the front-end terminal (http://44.217.21.102:3001/) and paste it in a browser. (e.g. Google Chrome)

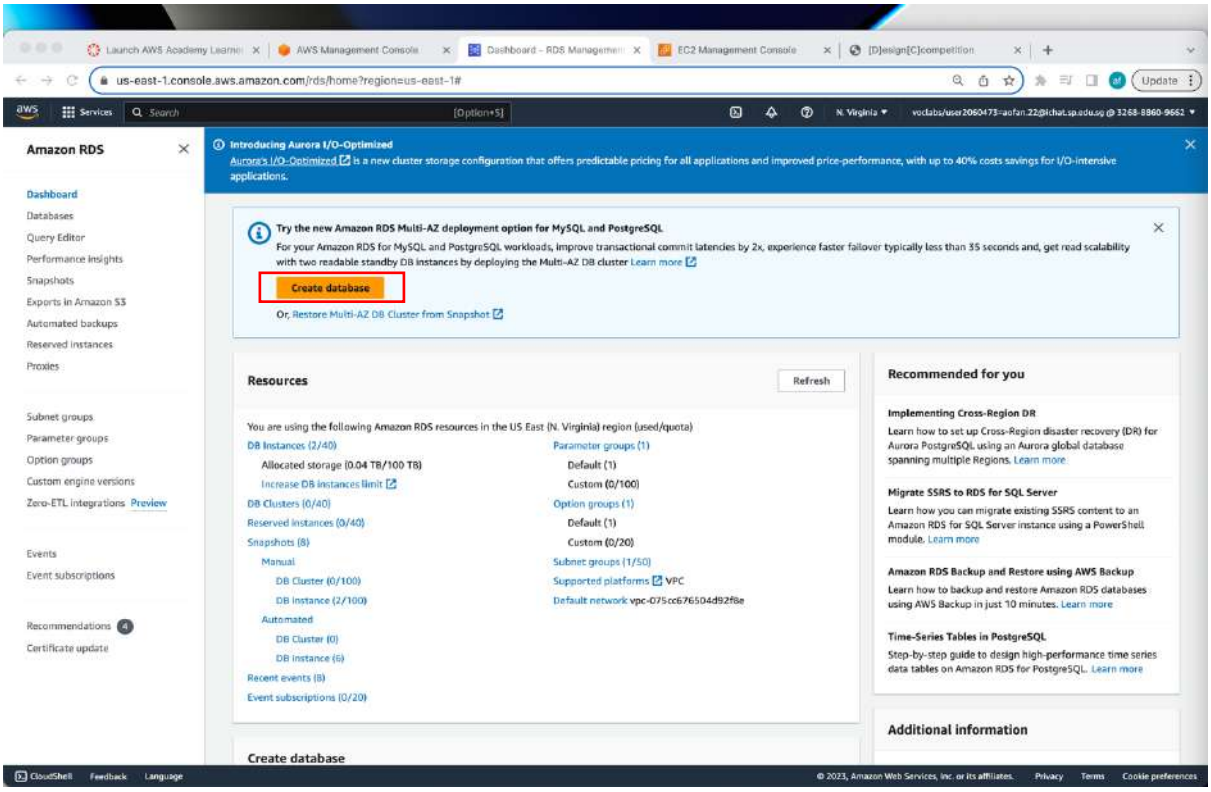


5. Setting up RDS database Instance

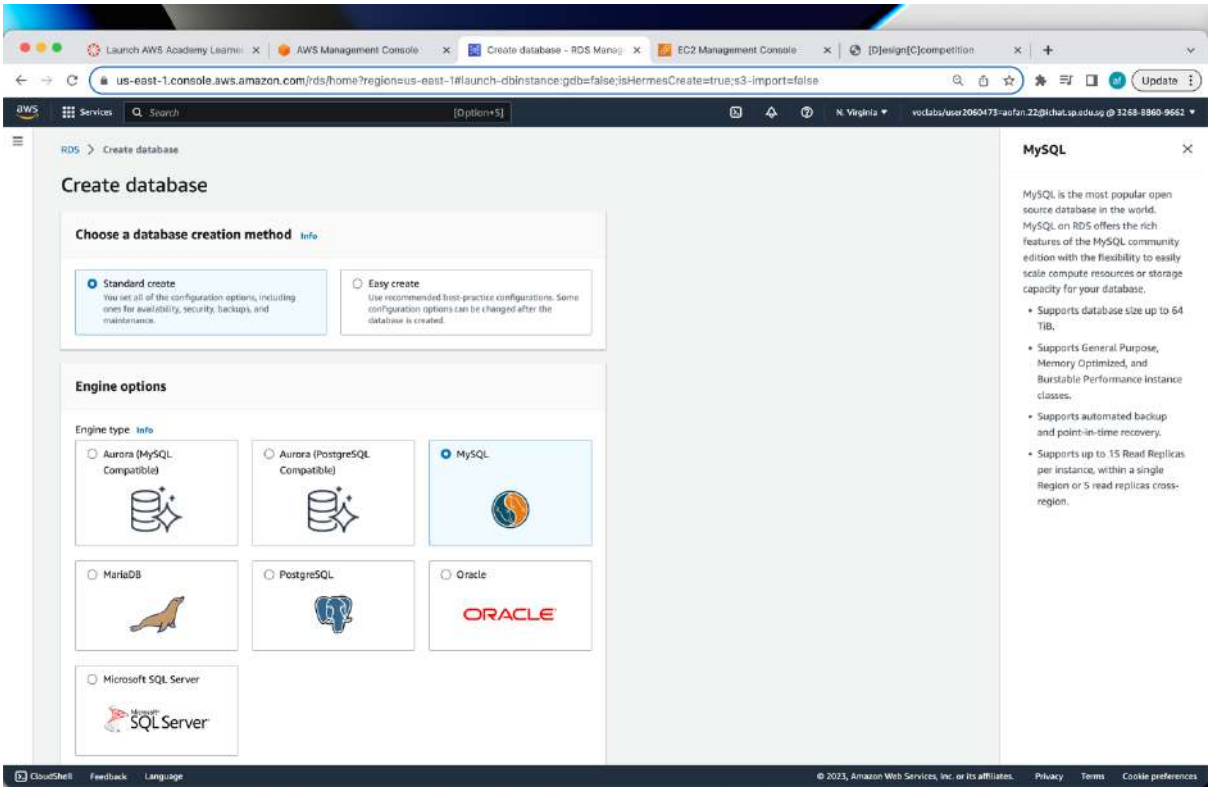
- Navigate to AWS Management Console and search RDS.



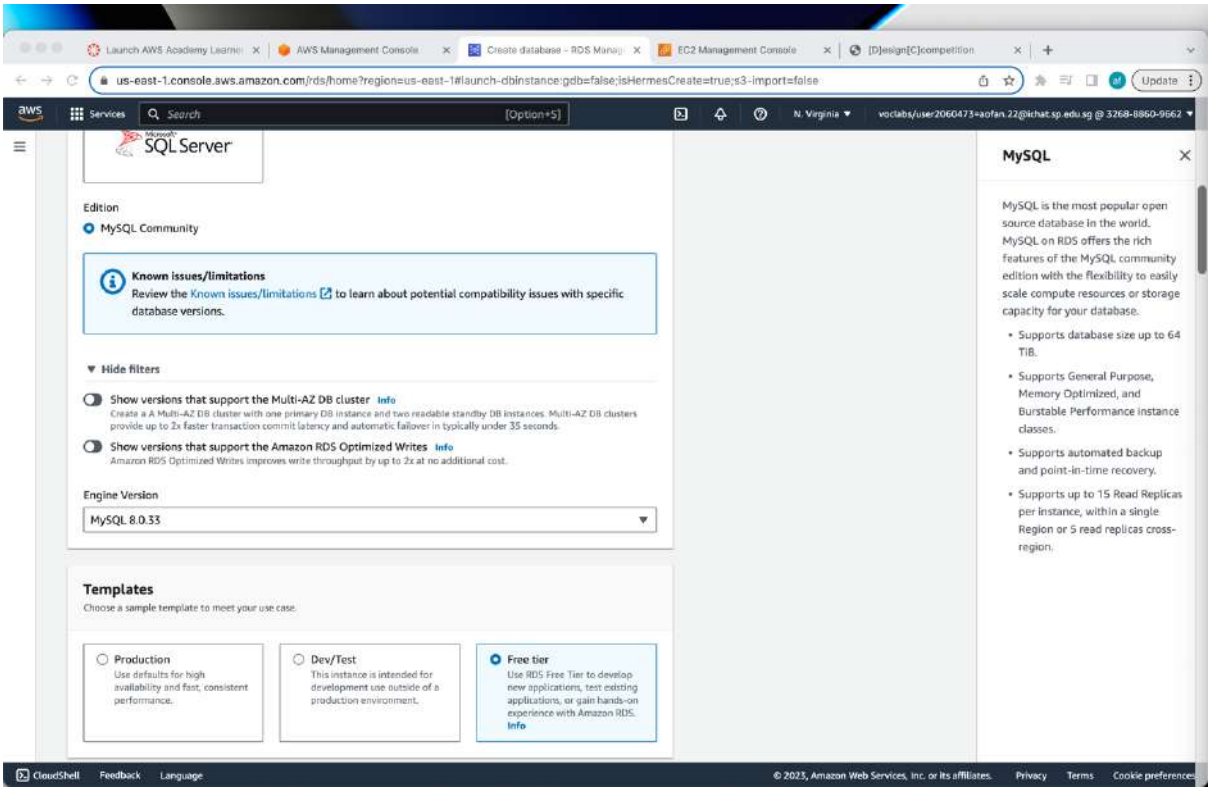
- In the RDS Dashboard, click Create database.



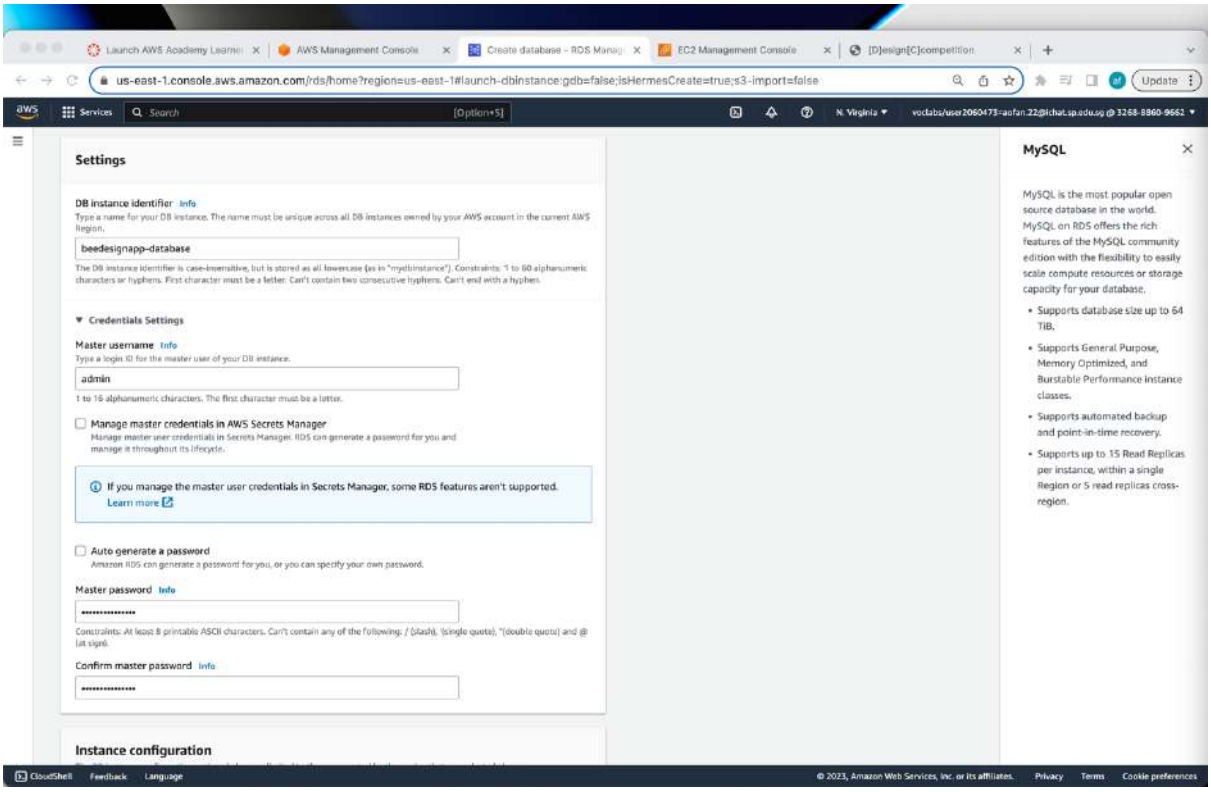
- Select Standard create under Choose a database creation method. Select MySQL under Engine options.



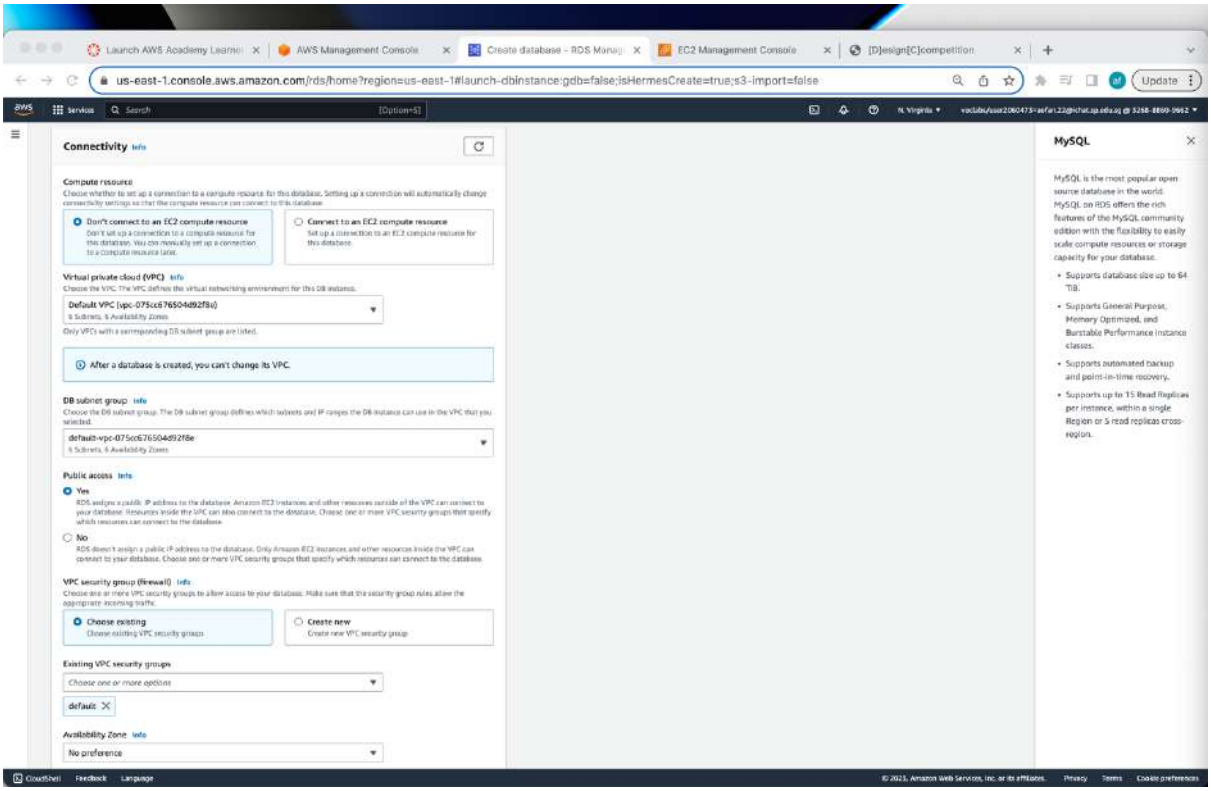
- Leave the Engine Version as default. Select Free tier under Templates.



- Under Settings, enter beedesignapp-database as DB instance identifier. Set Master password for the database.

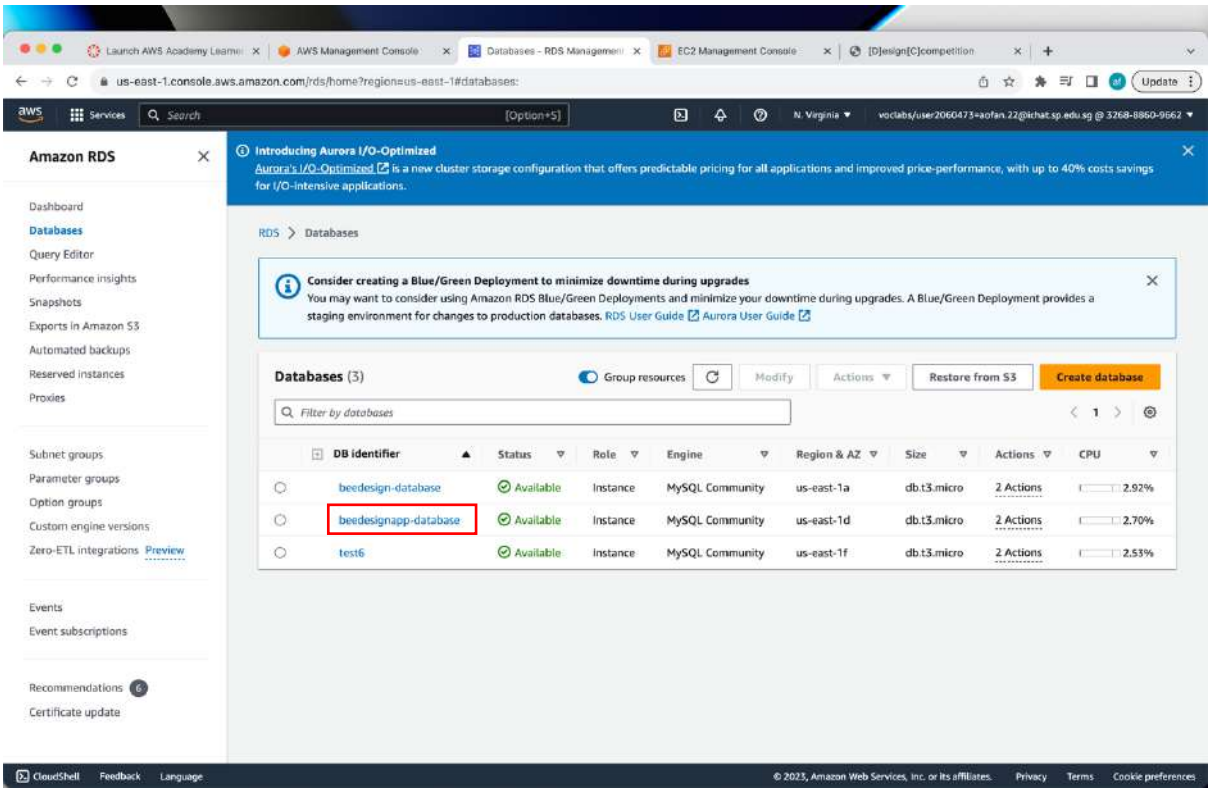


- Under Connectivity, select Yes for Public access.

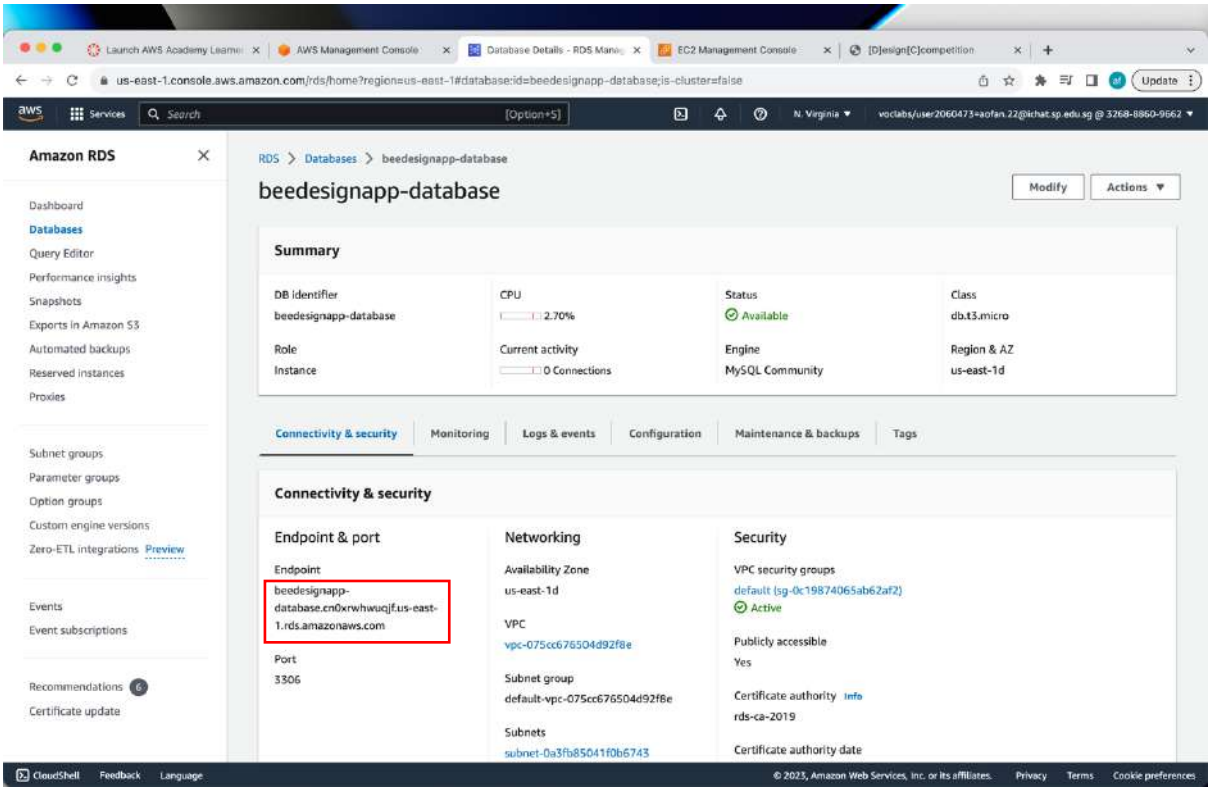


- You can leave rest of the settings as default and scroll all the way down and click on the Create database button.

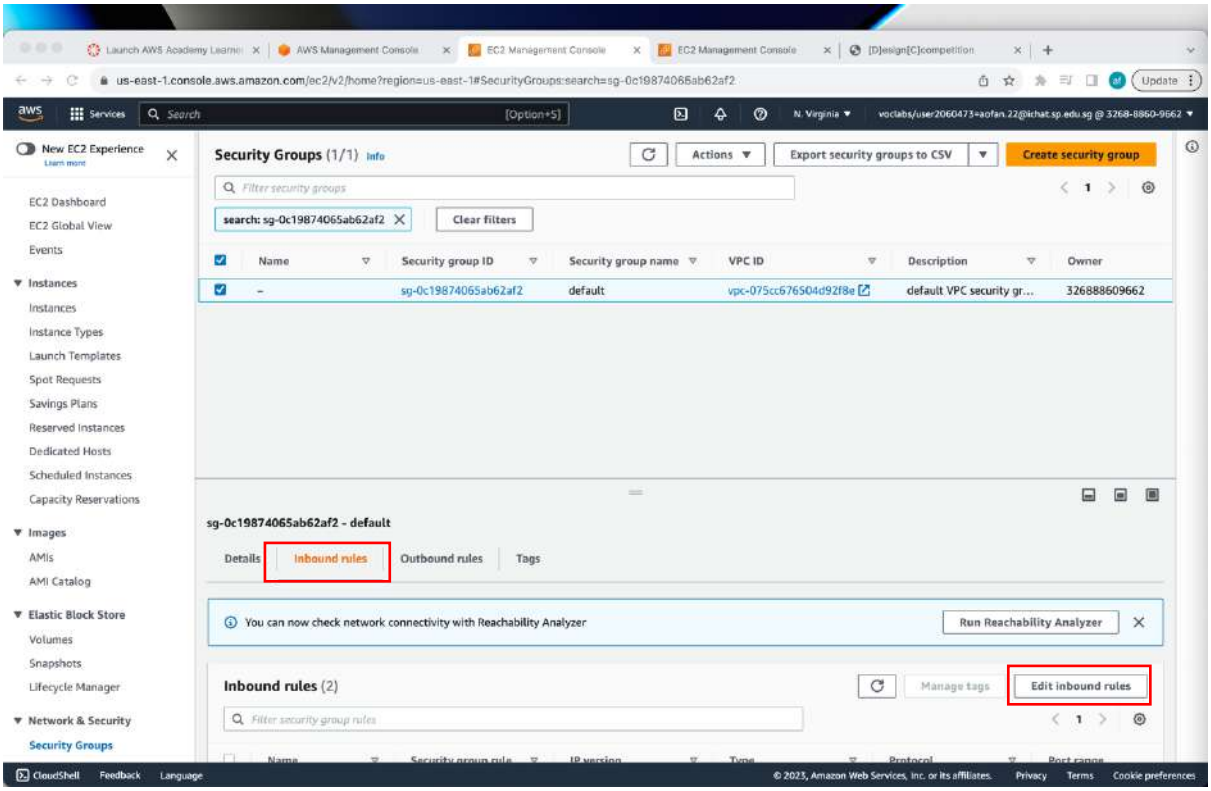
- Head to RDS Databases and selected the newly created database.



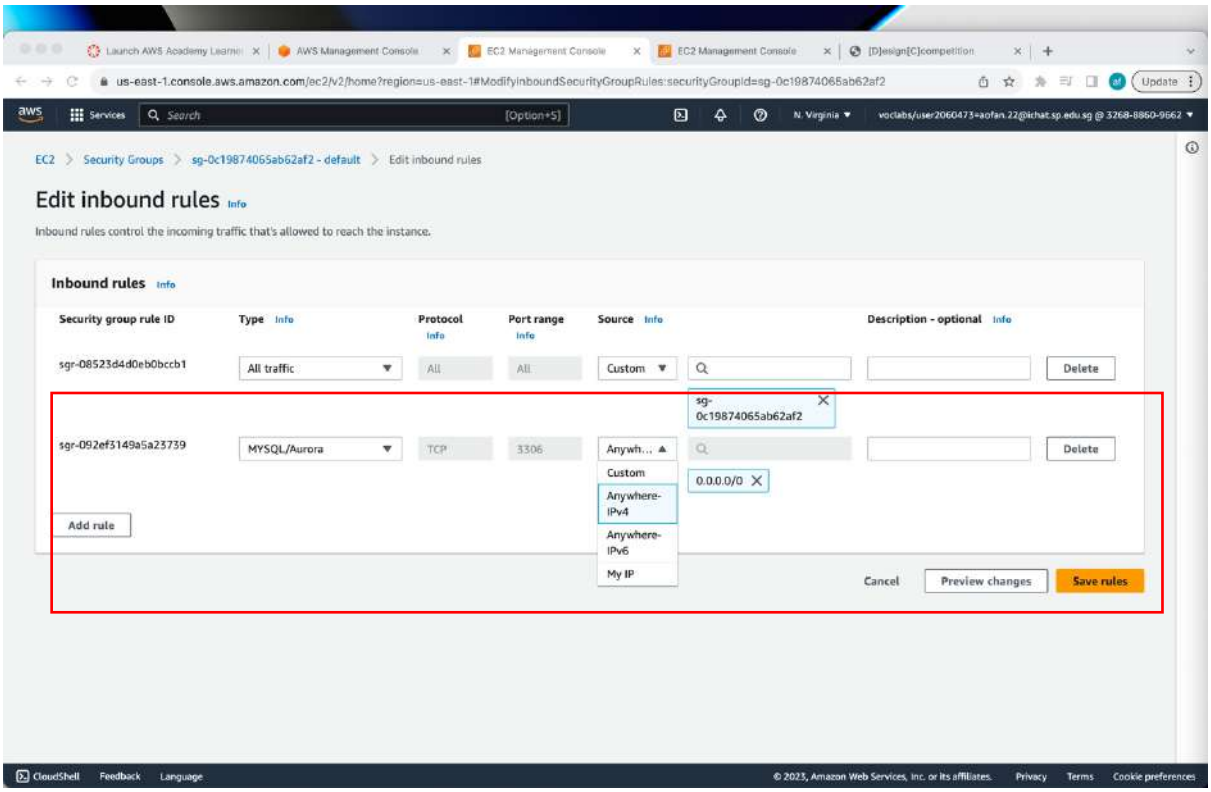
- In the database, copy the End point under Endpoint & port. Click on the link under VPC security groups.



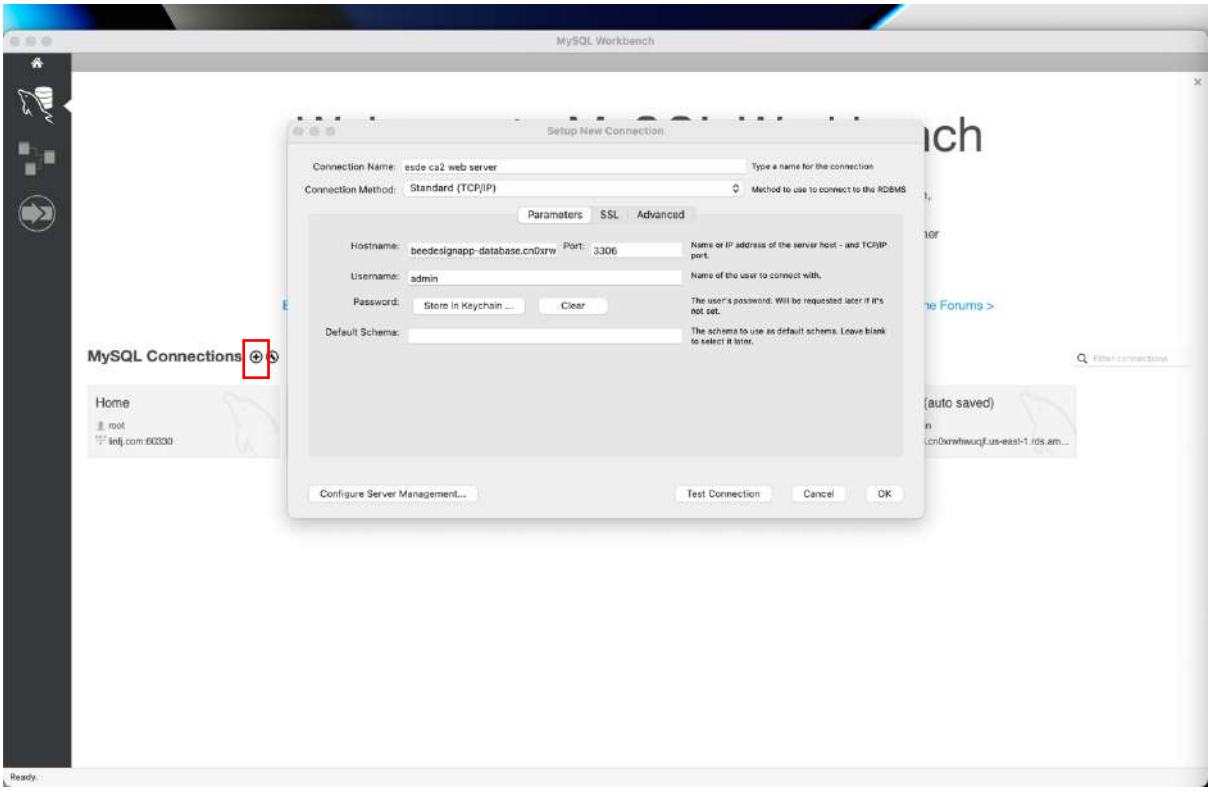
- Click on Inbound rules followed by Edit inbound rules



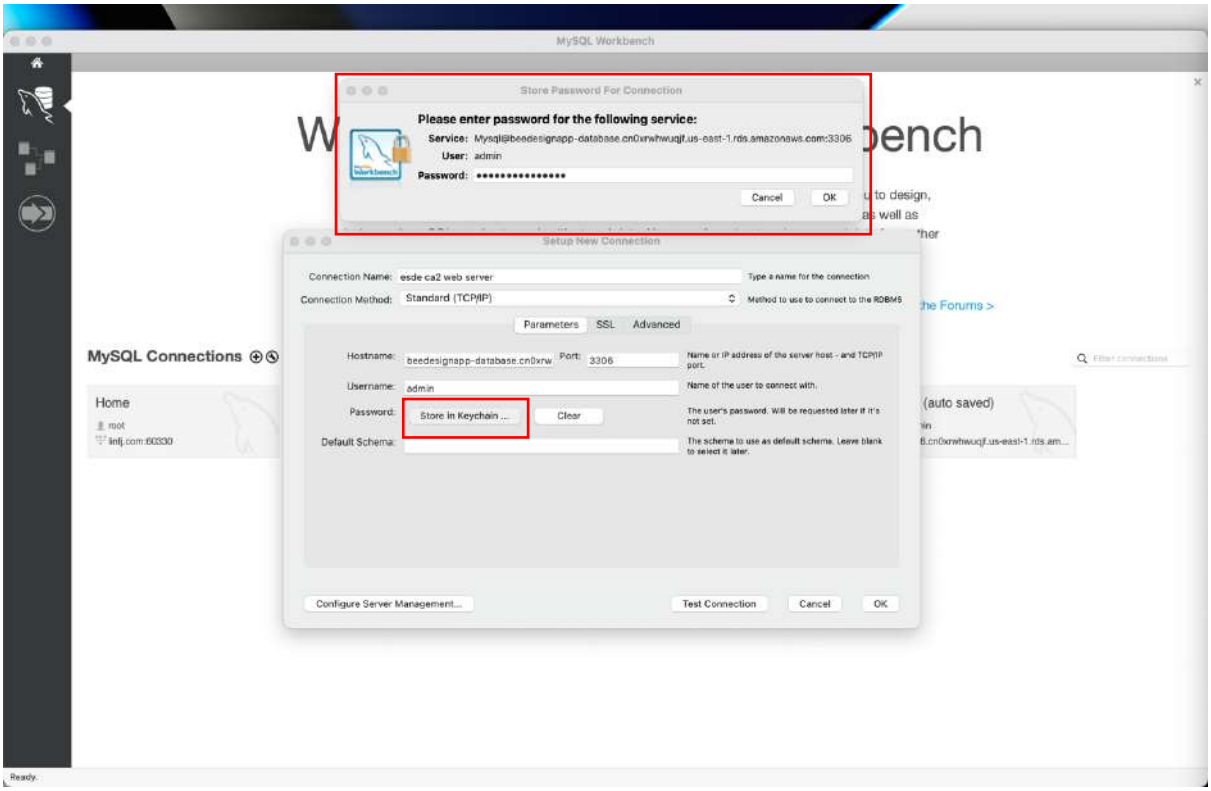
- Click Add rule, select MYSQL/Aurora under Type, select Anywhere-IPv4 under Source and click Save rules. (If you have the inbound rule for MYSQL you can skip this step)



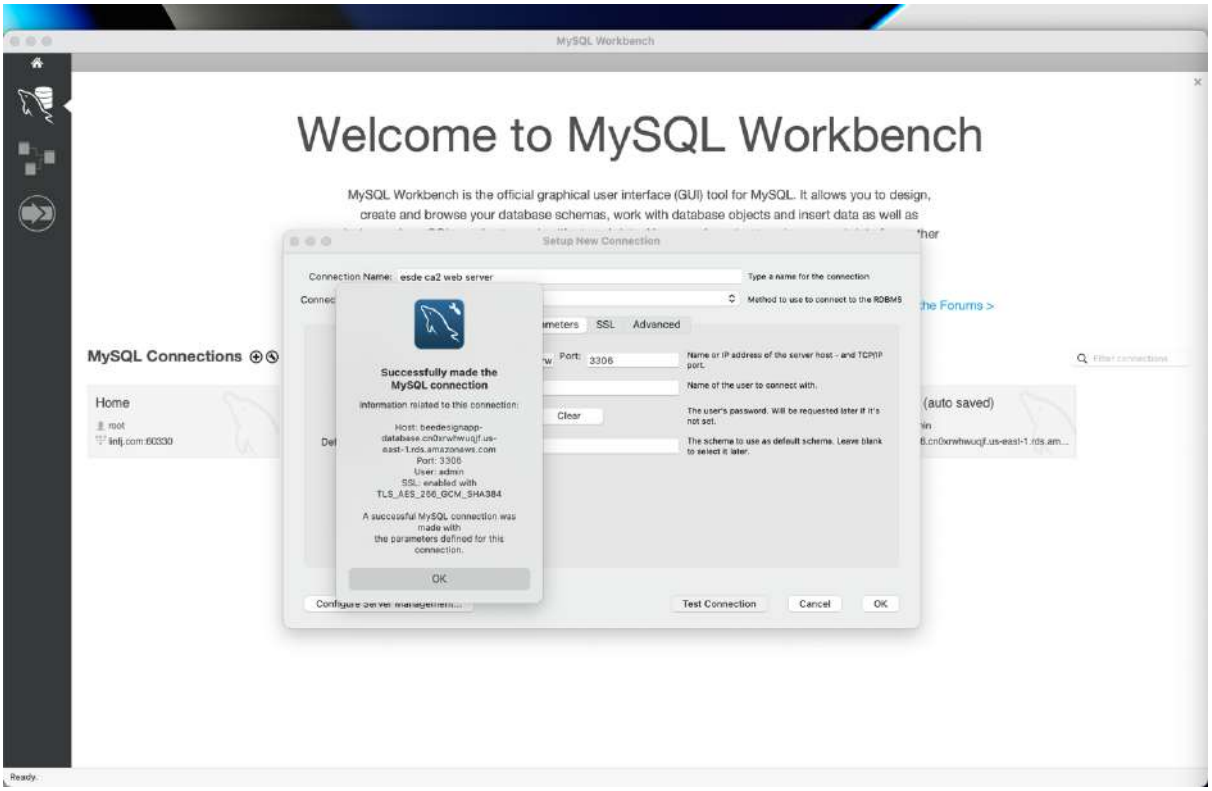
- Open MySQLWorkbench, click on the plus icon to setup a new connection. Enter esde ca2 web server in Connection Name:, paste the Endpoint of the RDS database in Hostname:, enter the Master username (admin) in Username:.



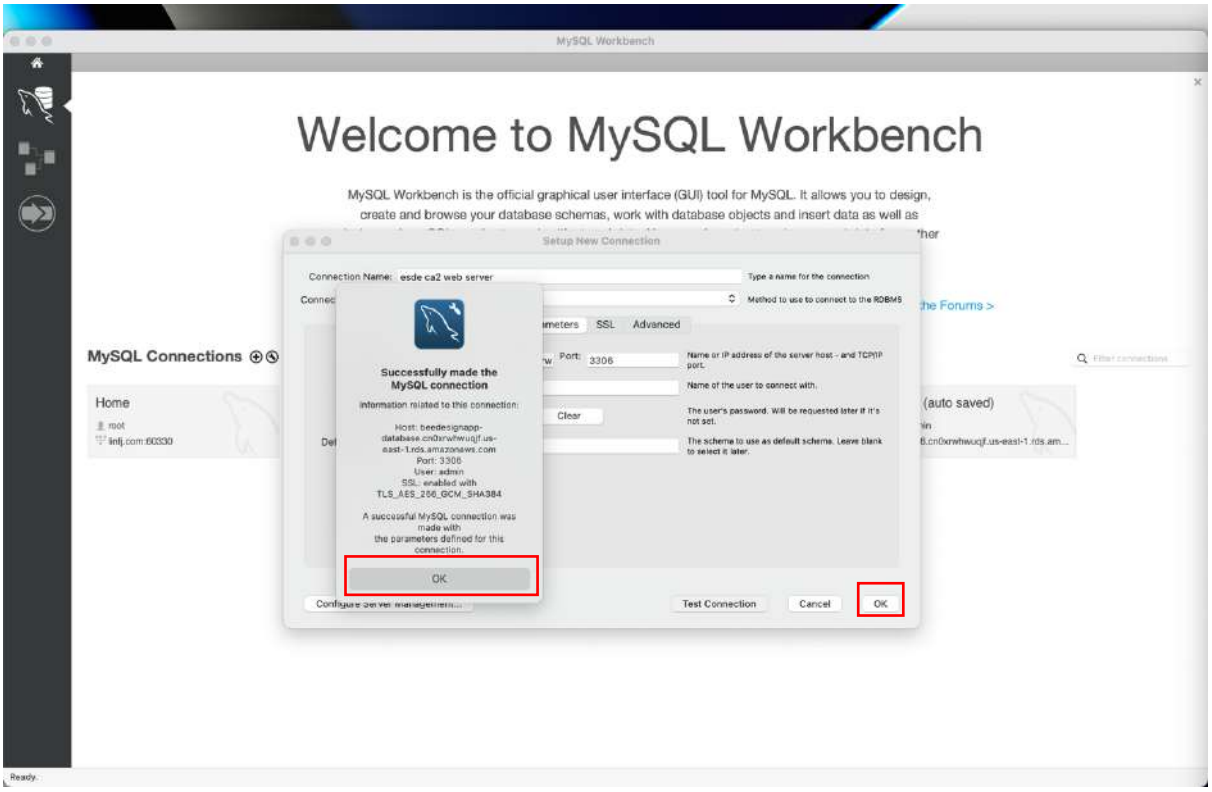
- Click on Store in Keychain .. and enter the Master password. (The password set when creating RDS database)



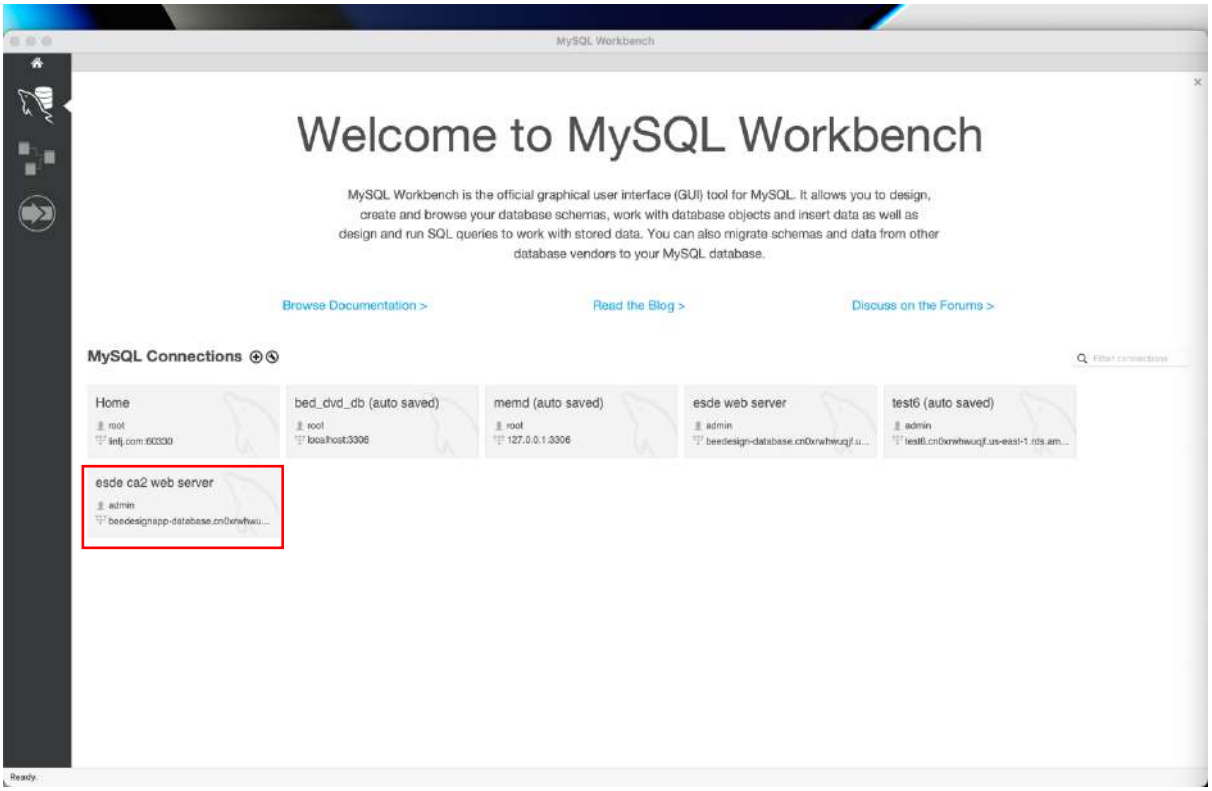
- Click Test Connection to verified that the connection can be established.



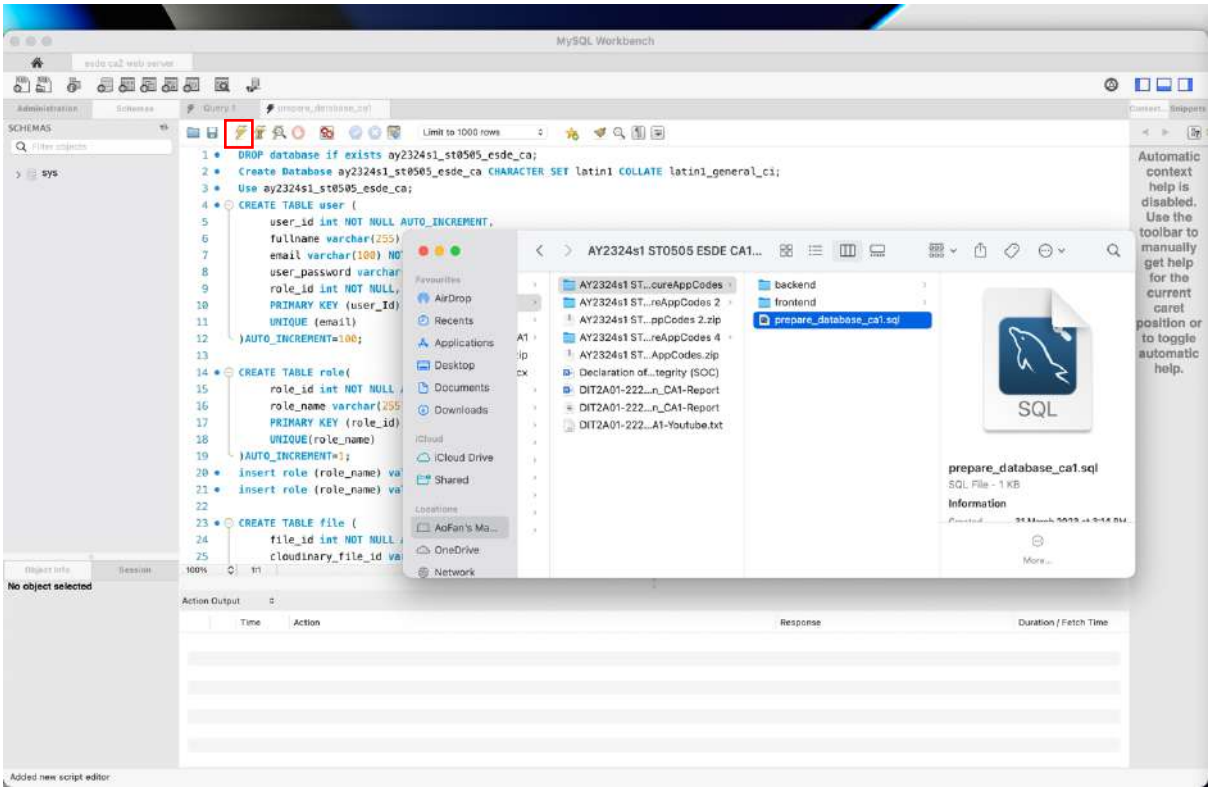
- Click OK to close the Test Connection result and click the OK button to establish the connection.



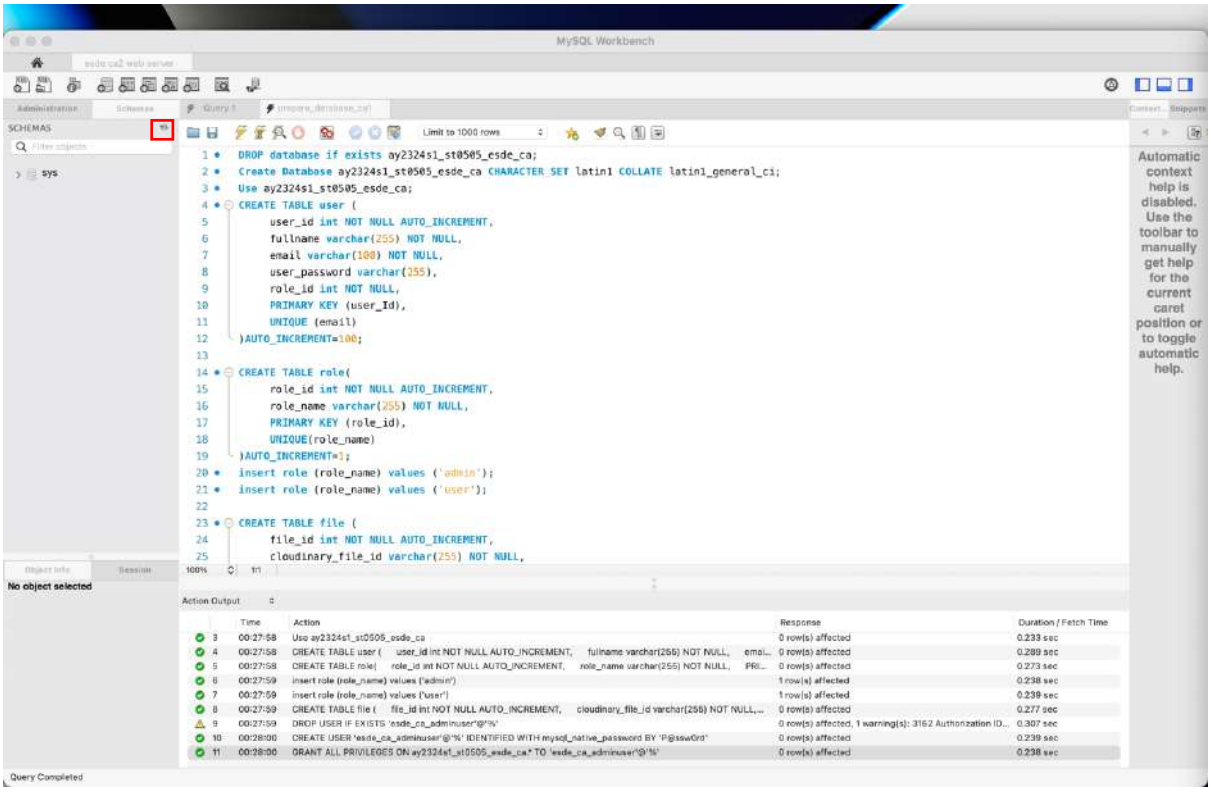
- Click on the connection.

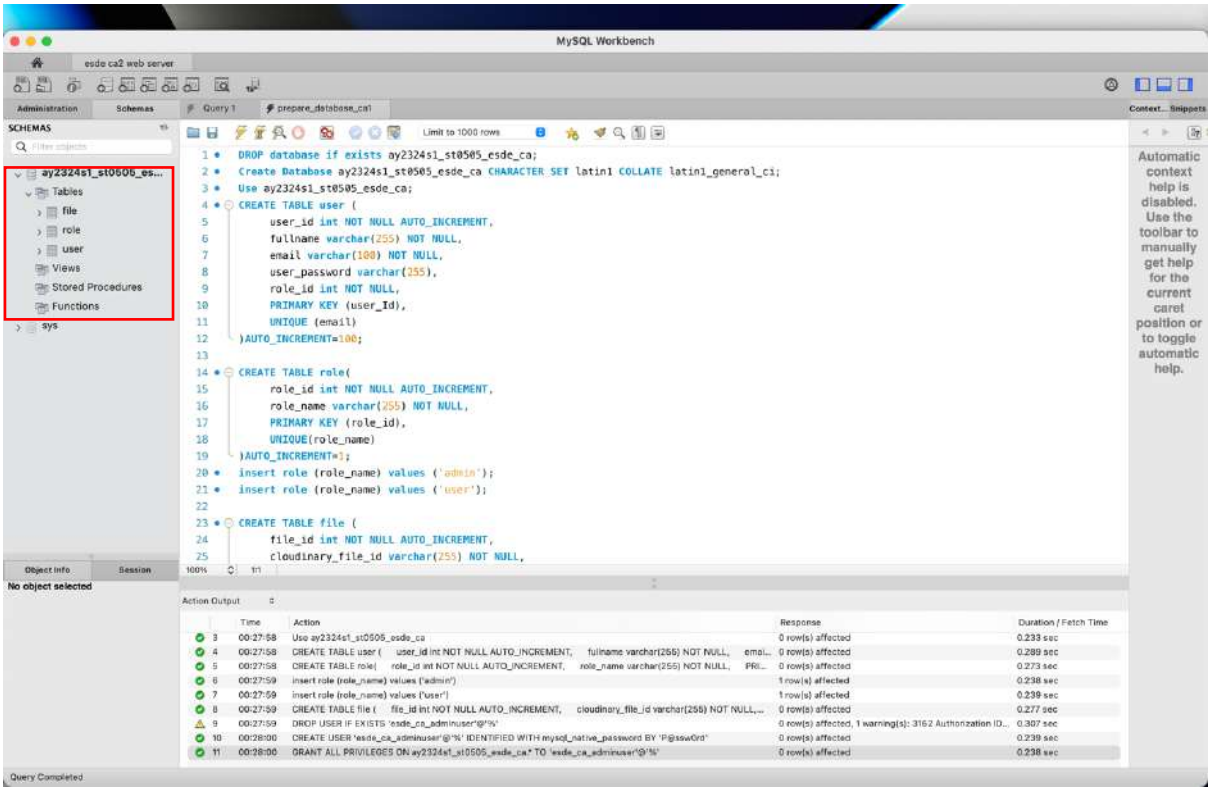


- Open prepare_database_ca1 file to prepare the database. Click on the lightning button to execute the codes.

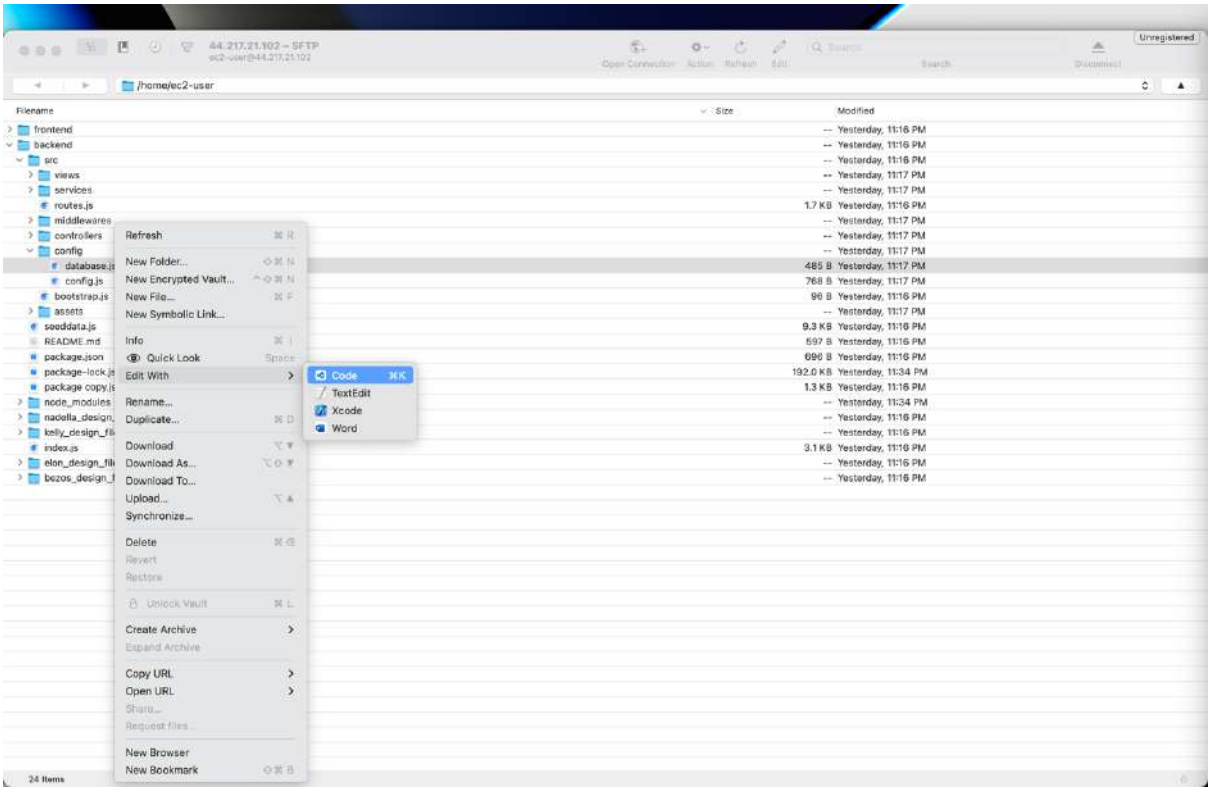


- Refresh the schemas to see the result.

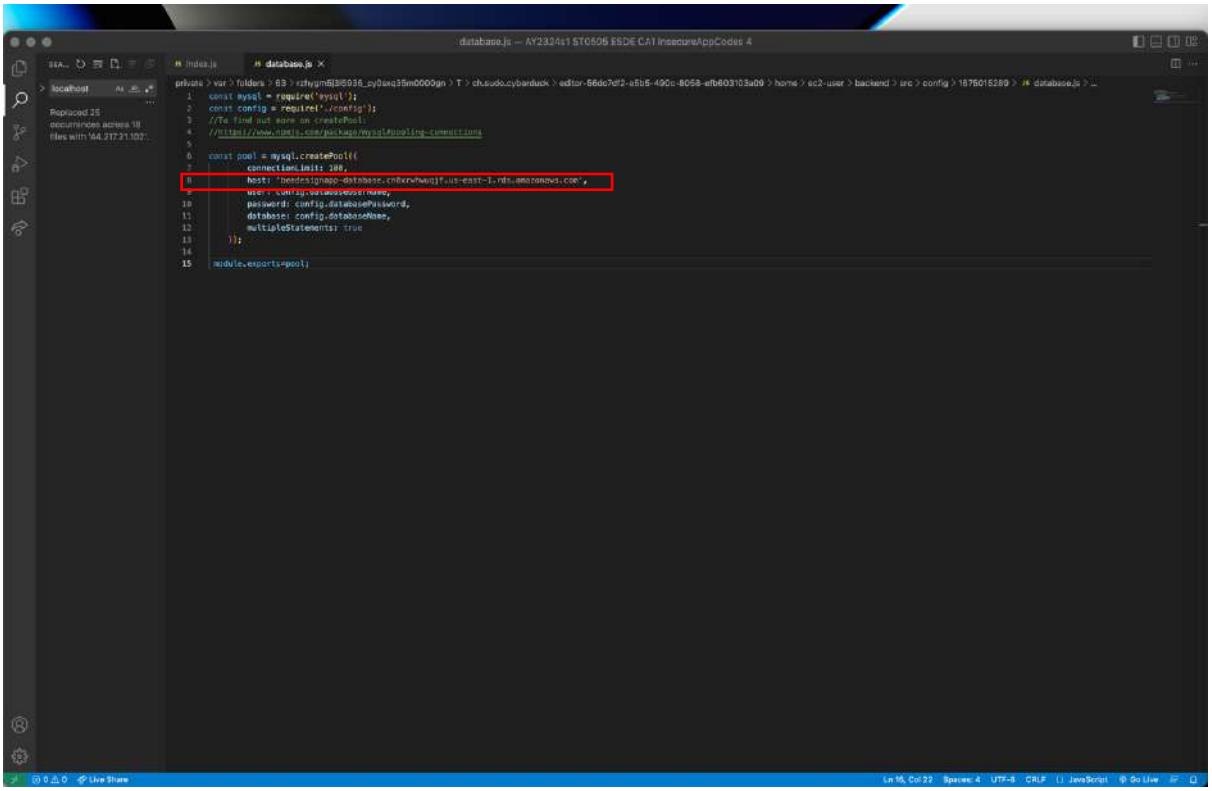




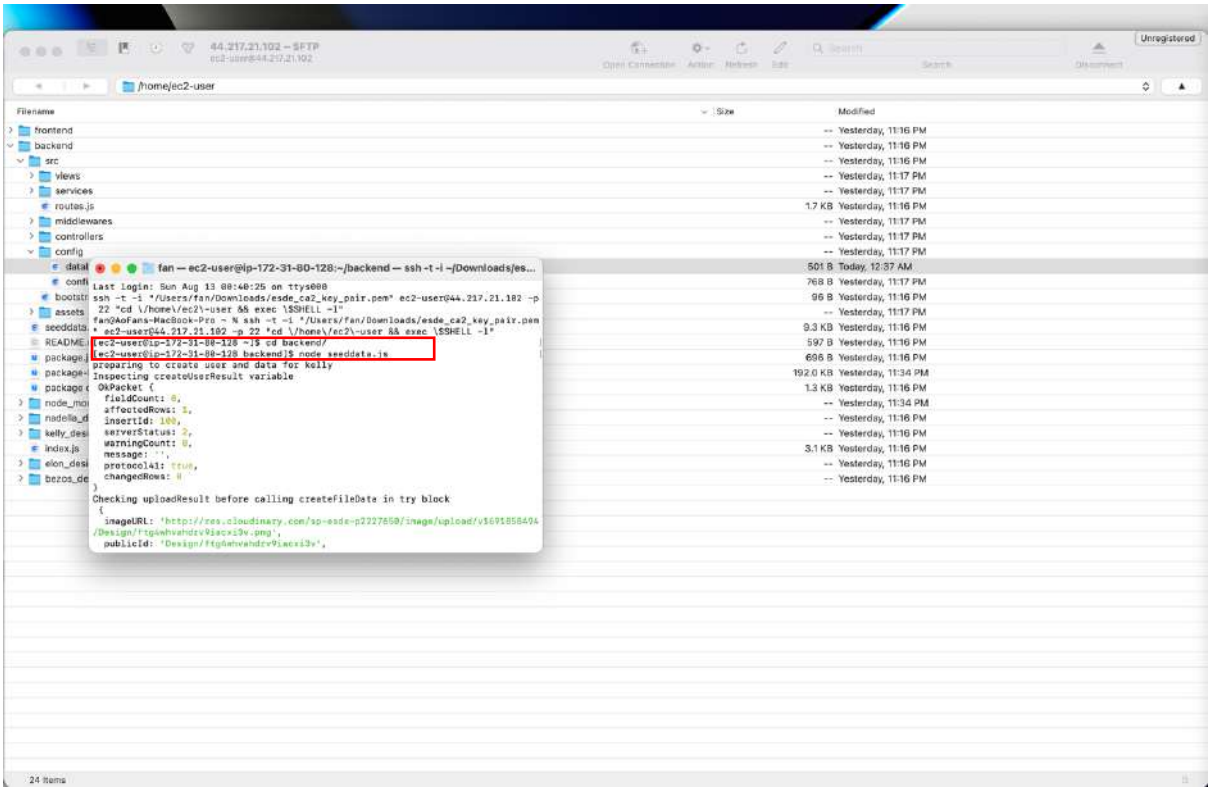
- Go to Cyberduck, right click database.js located in backend → src → config. Hover over Edith with and select Code.



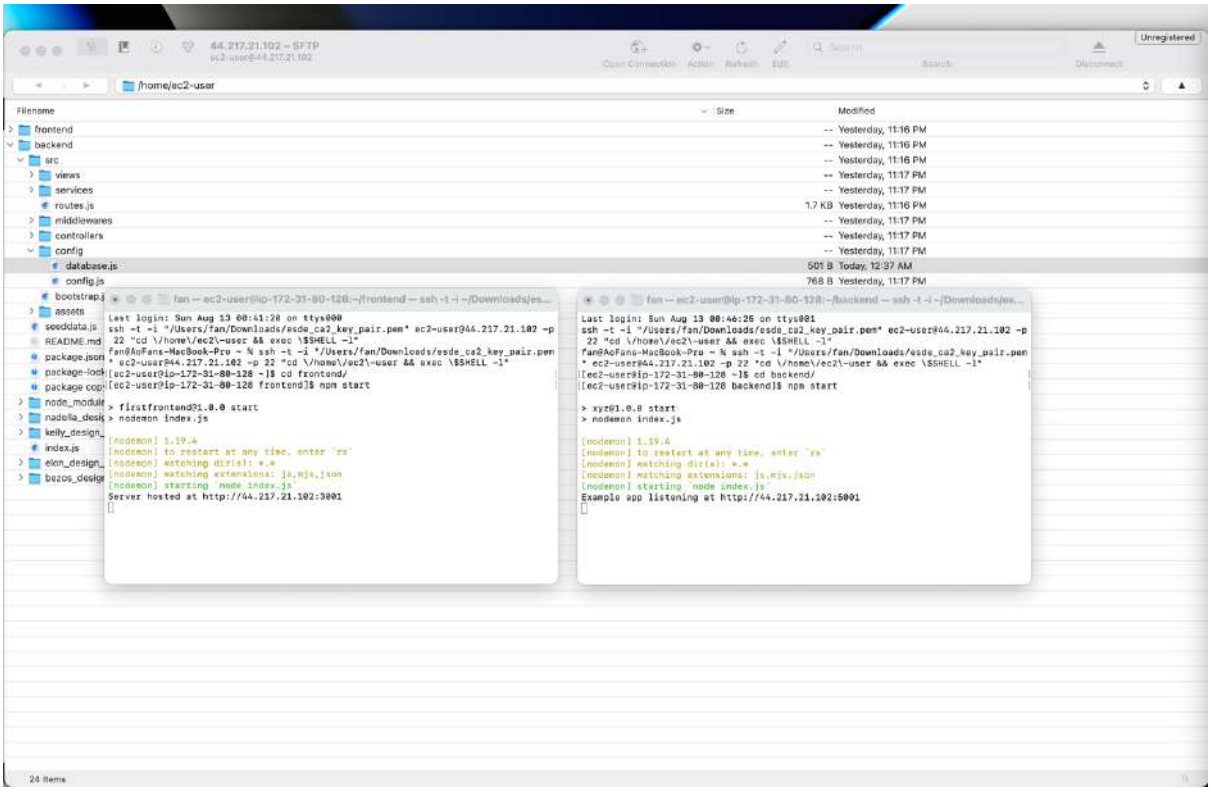
- In Visual Studio Code, replace host with RDS Endpoint. (line 8)



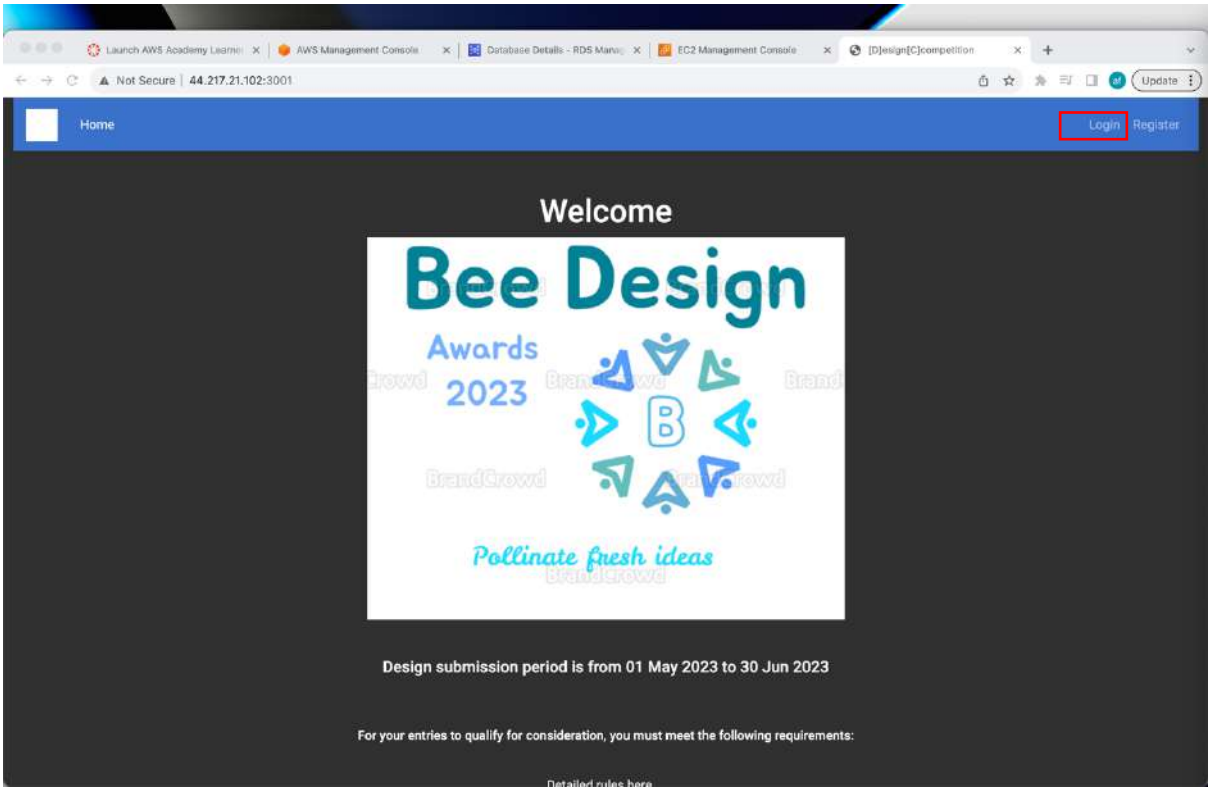
- Go to Cyberduck, open a terminal. cd to backend and run node seeddata.js.

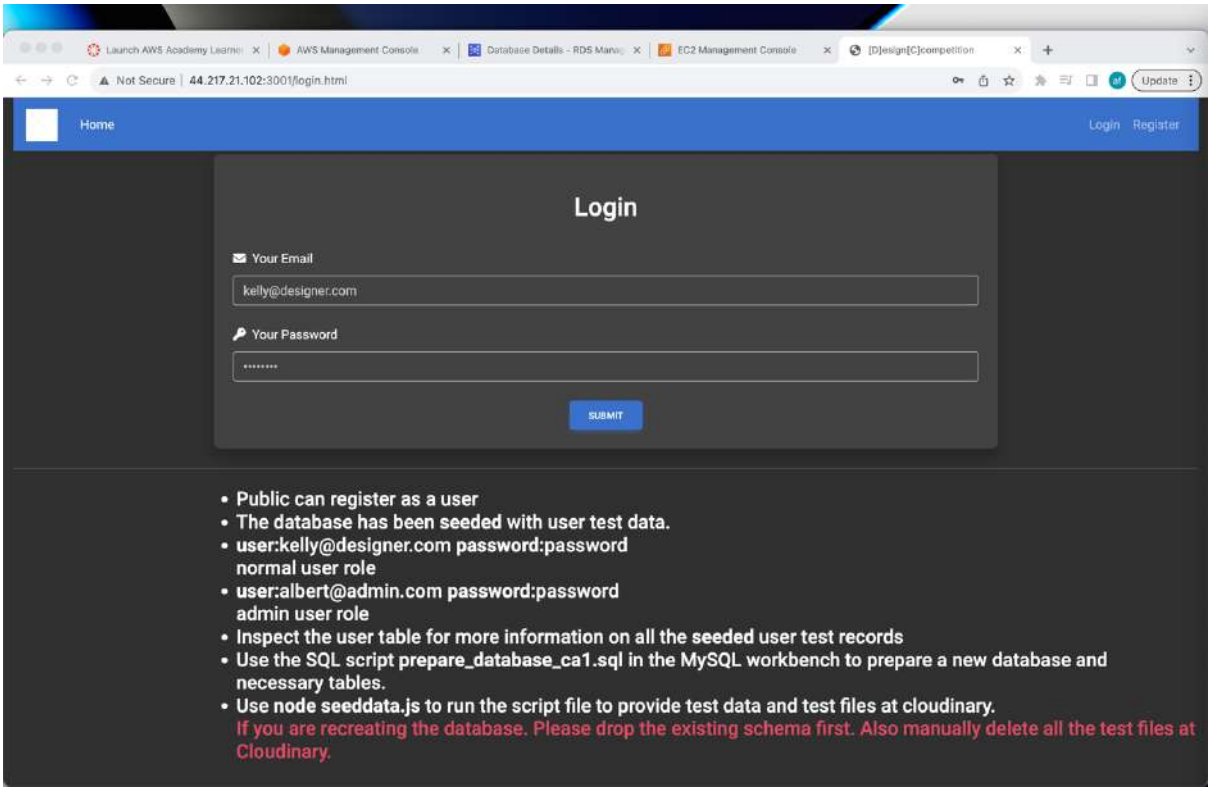


- Open 2 terminals in Cyberduck, cd the front-end and back-end respectively and run npm start.

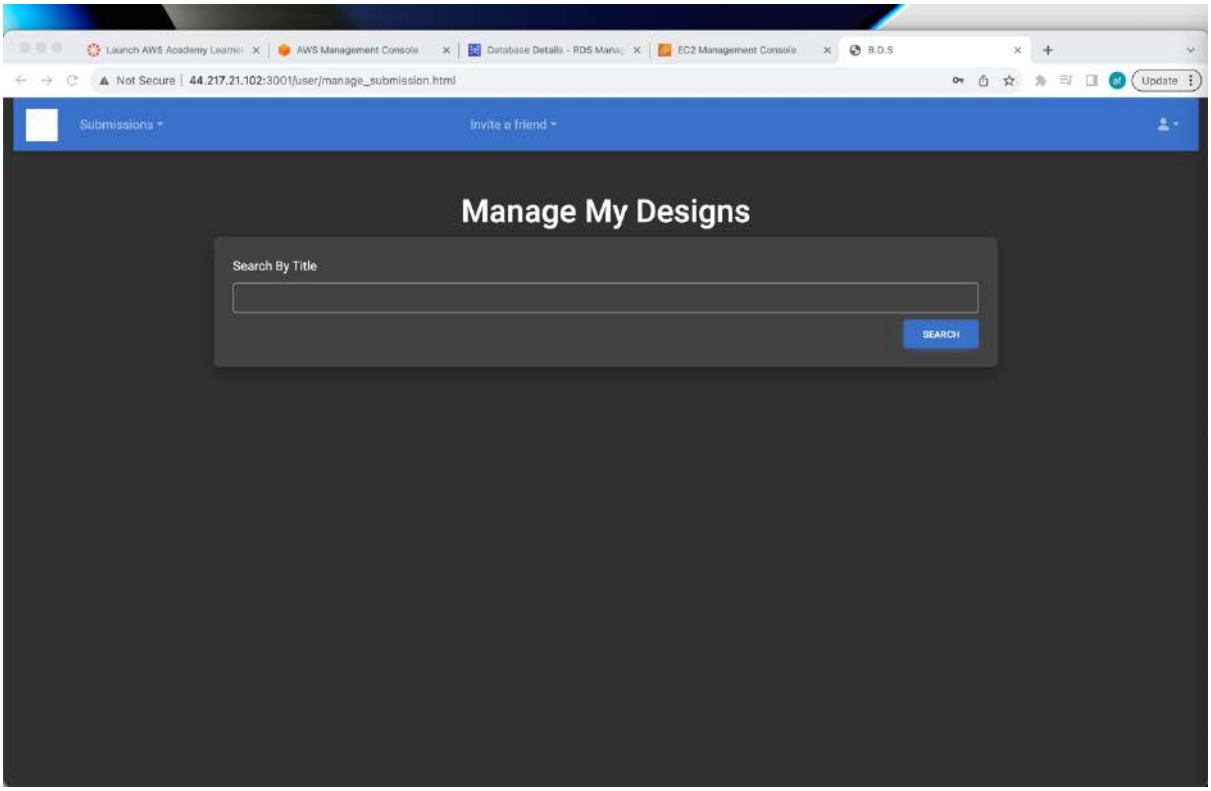


- Open the bee design app on browser and click on Login. Enter kelly@designer.com in Your Email and password in Your Password. Click on the SUBMIT button.



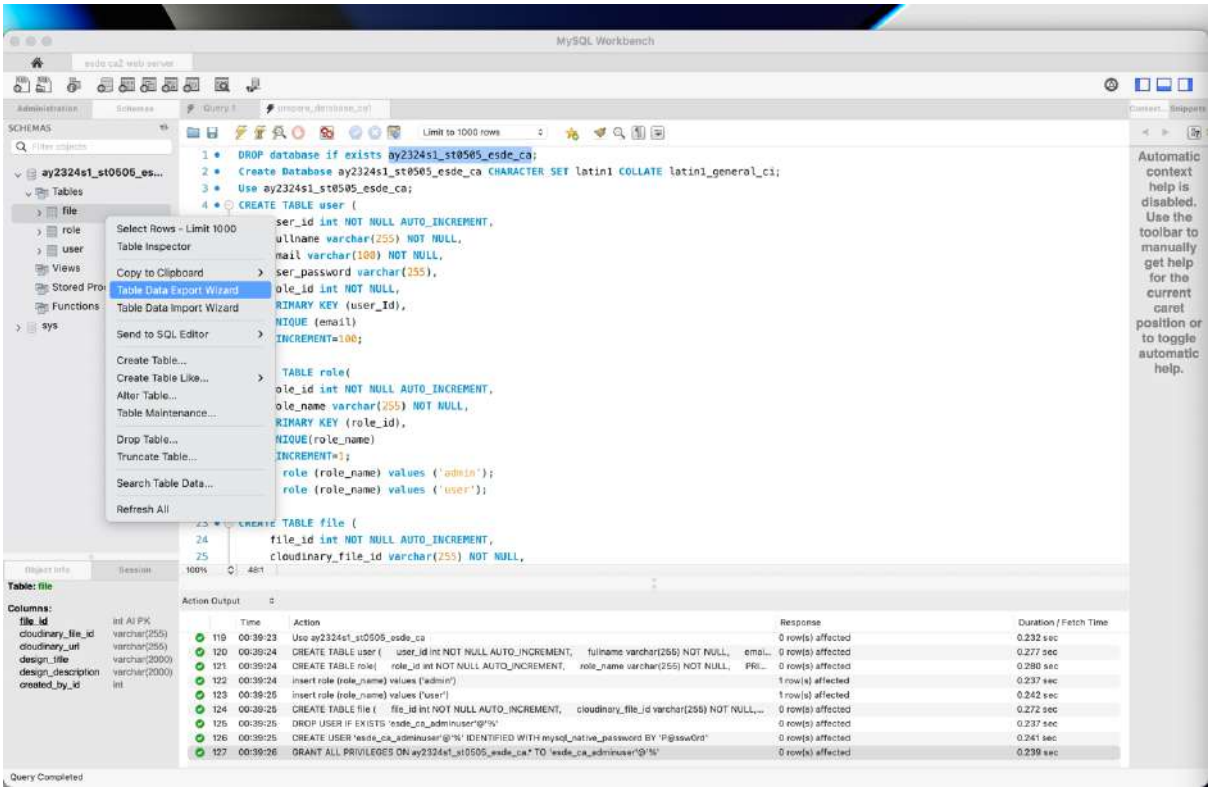


- When successfully logged in, it indicates that the bee design app is connected to the RDS database instance.

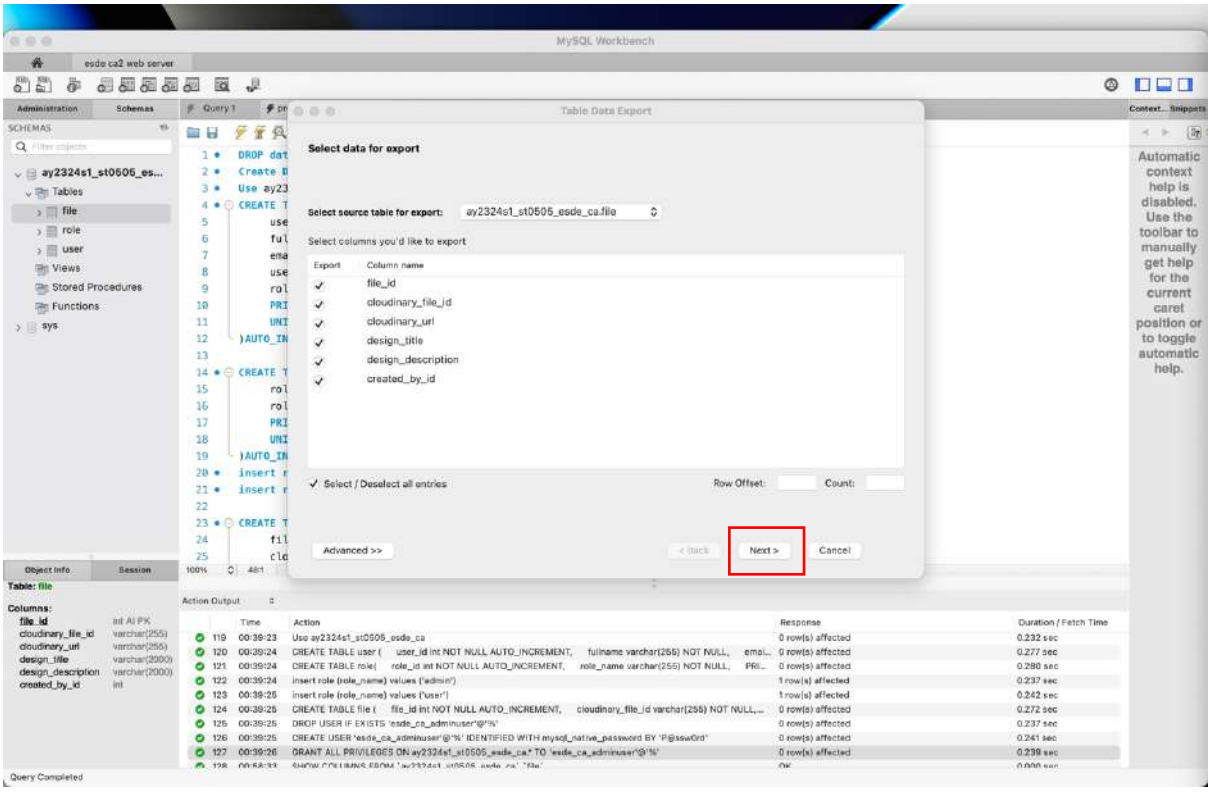


6. Creating and Populating DynamoDB table

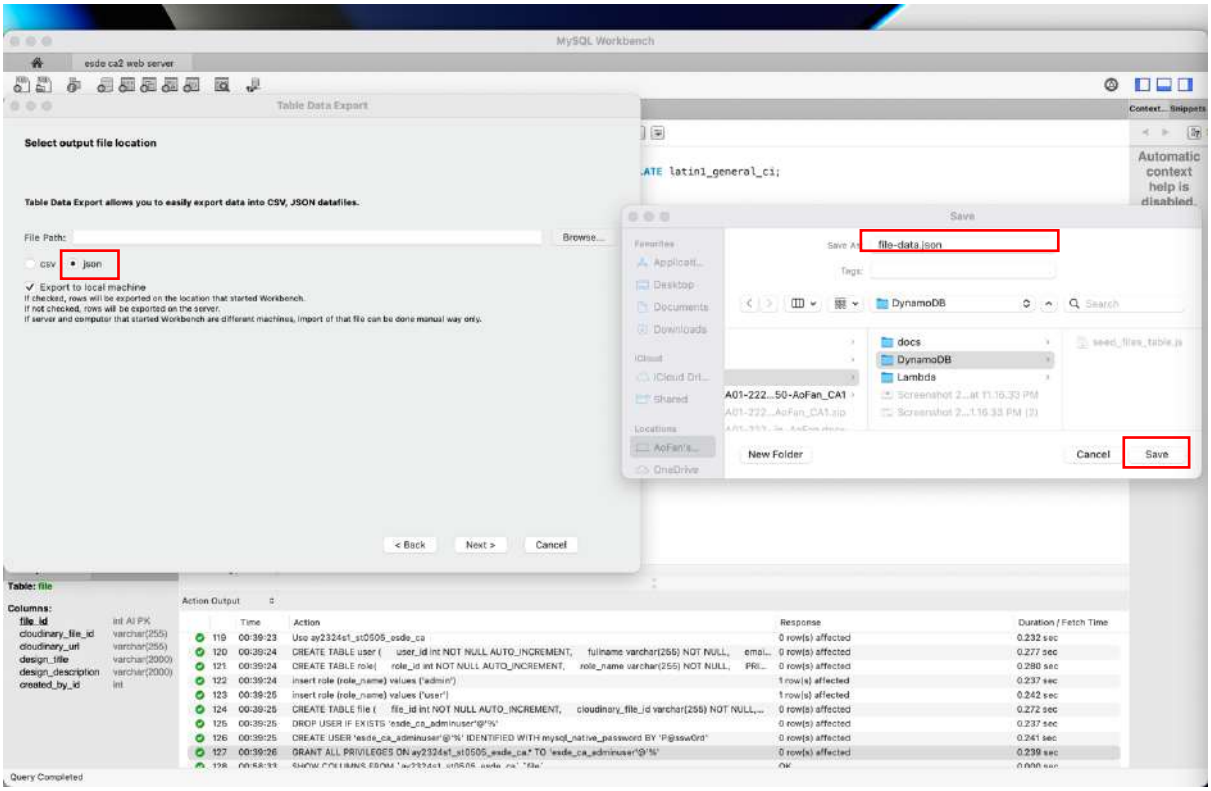
- Go to MySQLWorkbench, click on the esde ca2 web server connection. Right click on file located under ay2324s1_st0505_esde_ca, under Tables and select Data Table Export Wizard.



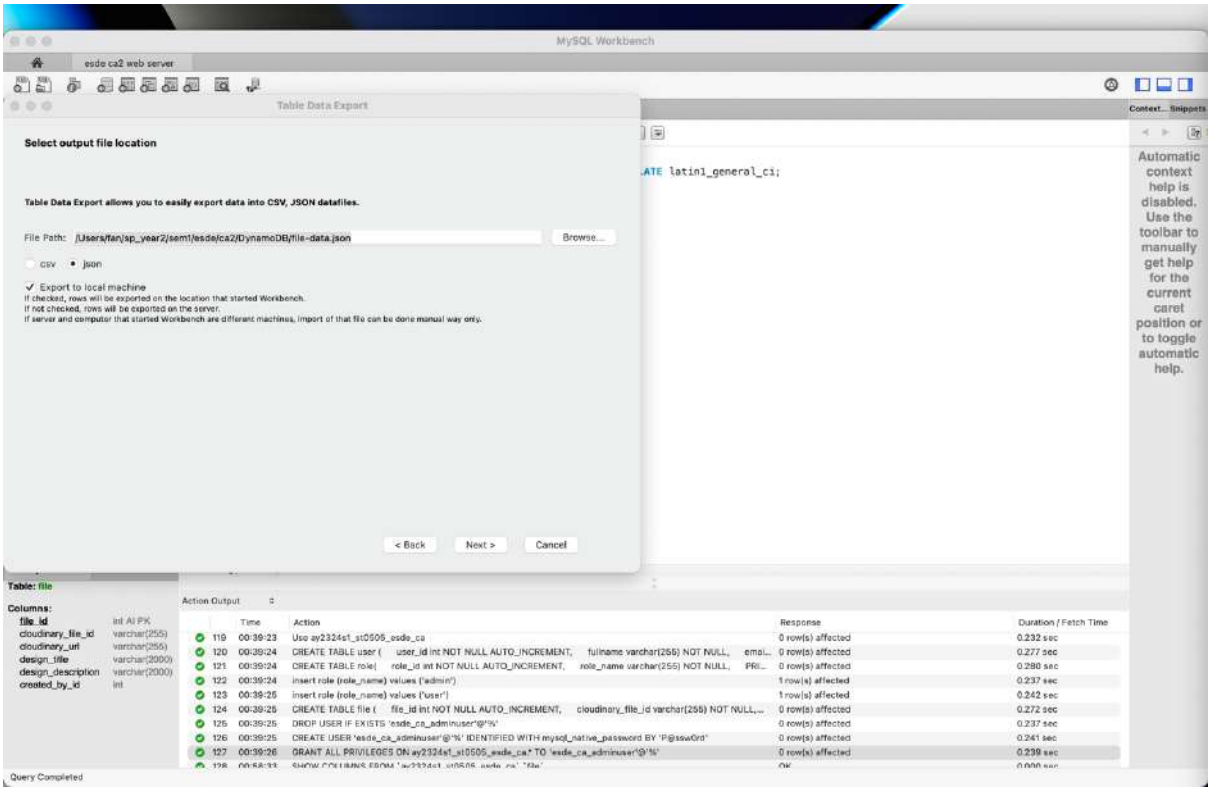
- On Table Data Export tab, click Next >.

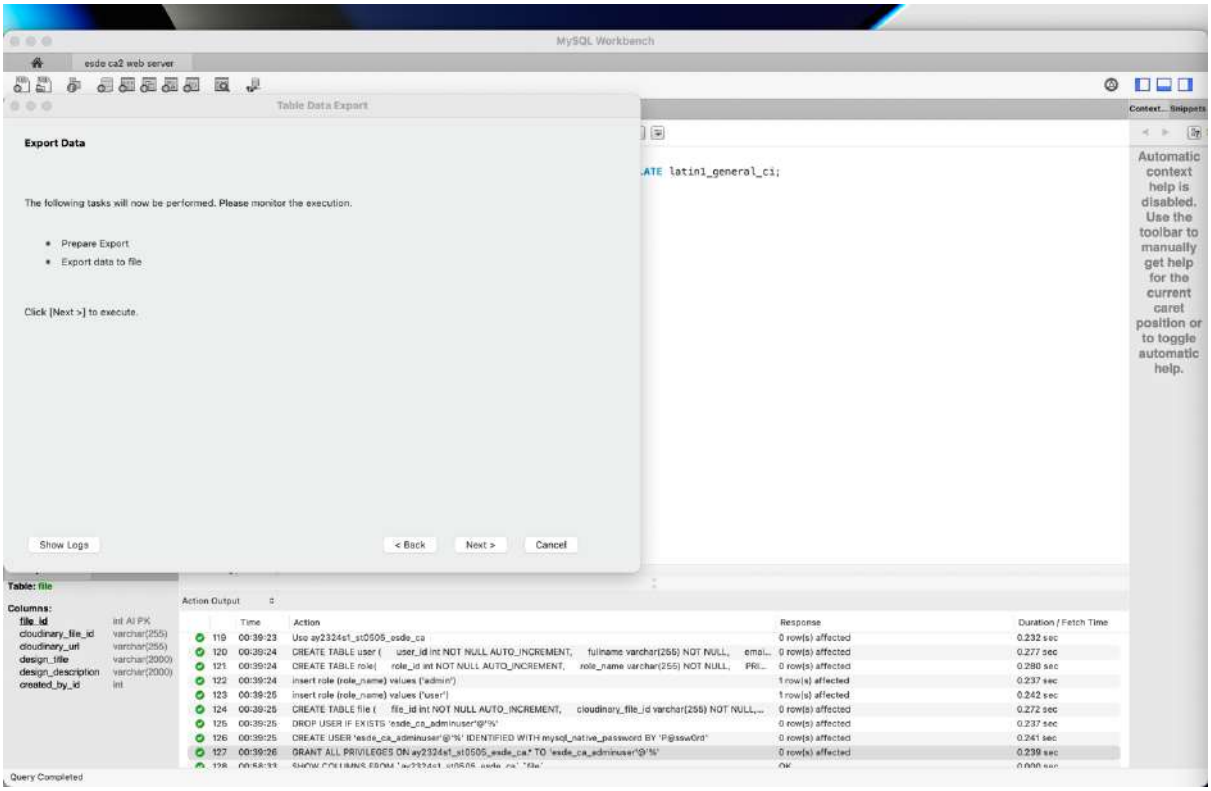


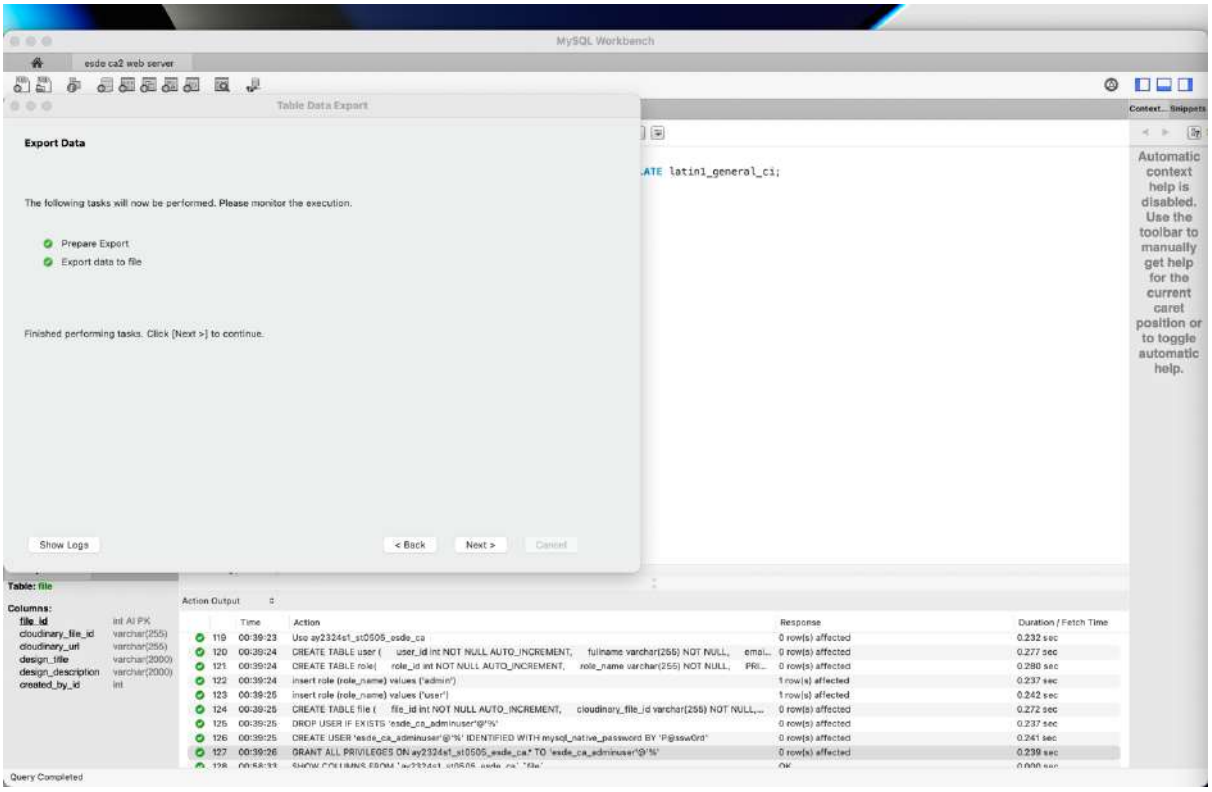
- Select json radio button and click on Browse… Name the file file-data.json and click Save.

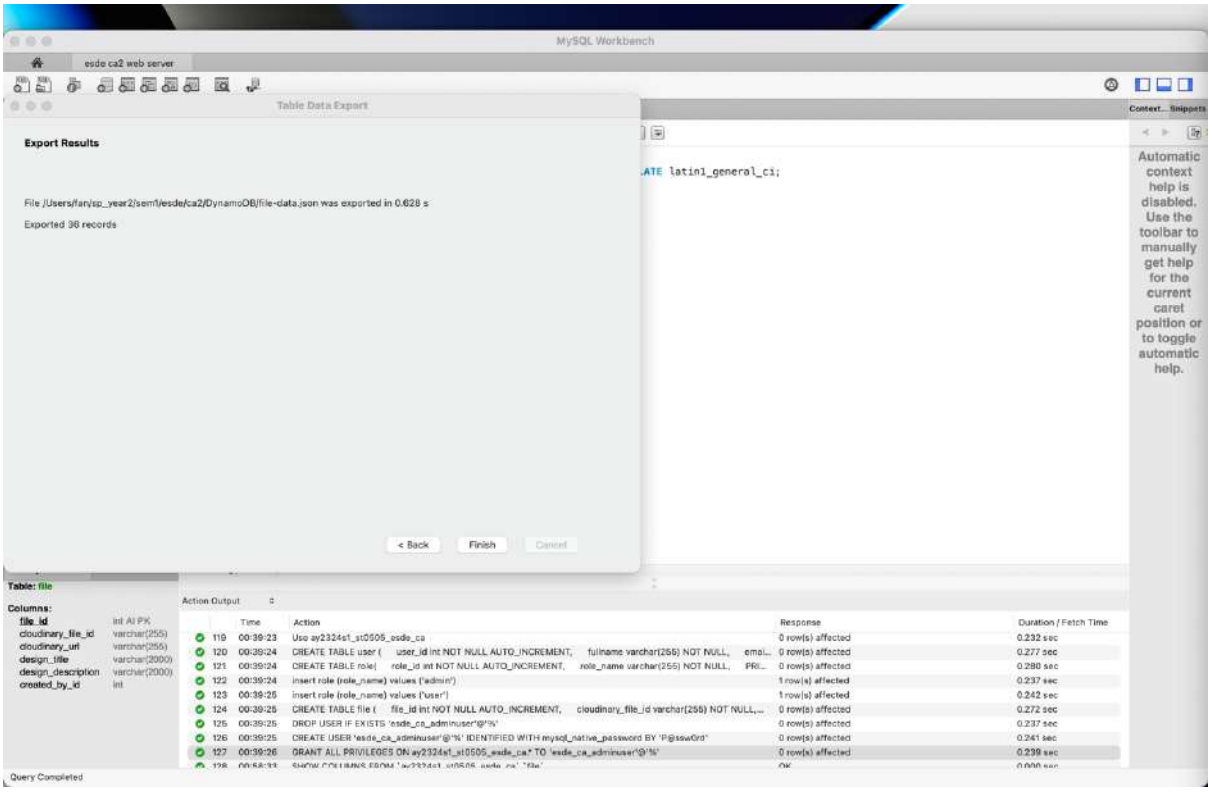


- Click Next > followed by Next > followed by Next >. Lastly Click Finish.

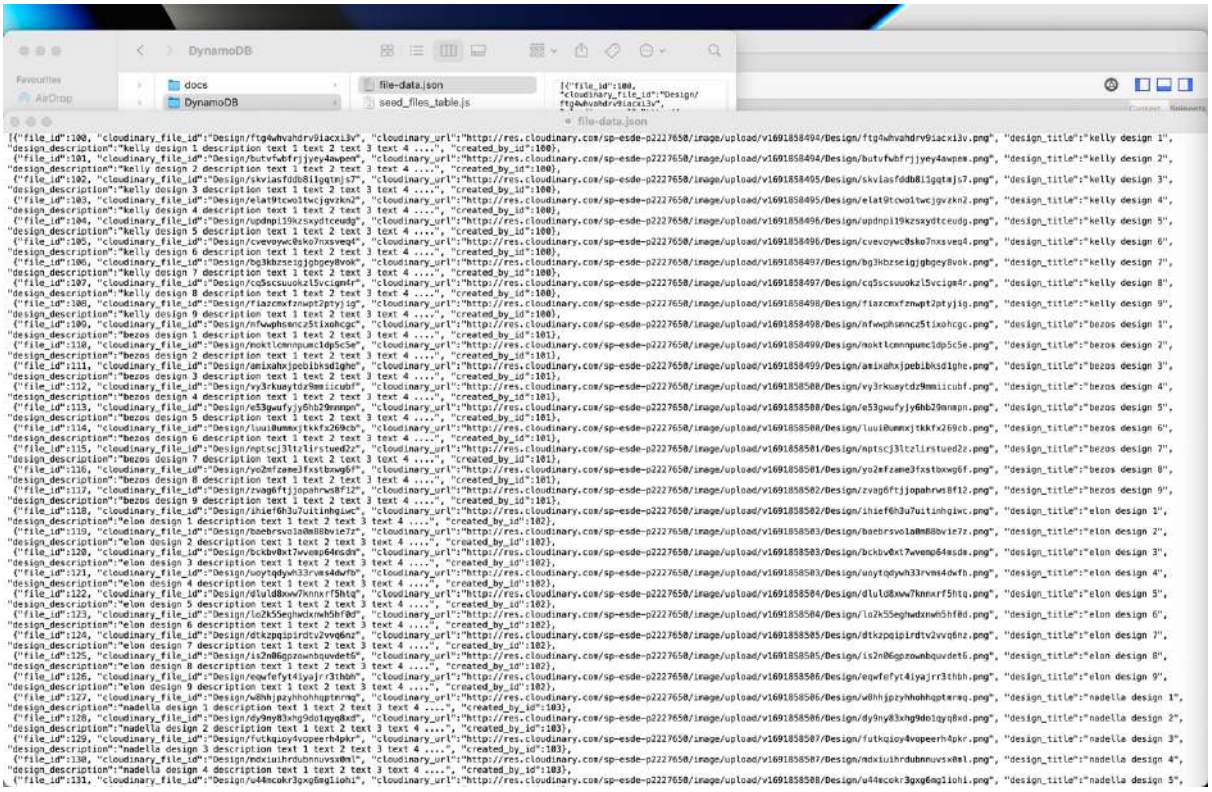




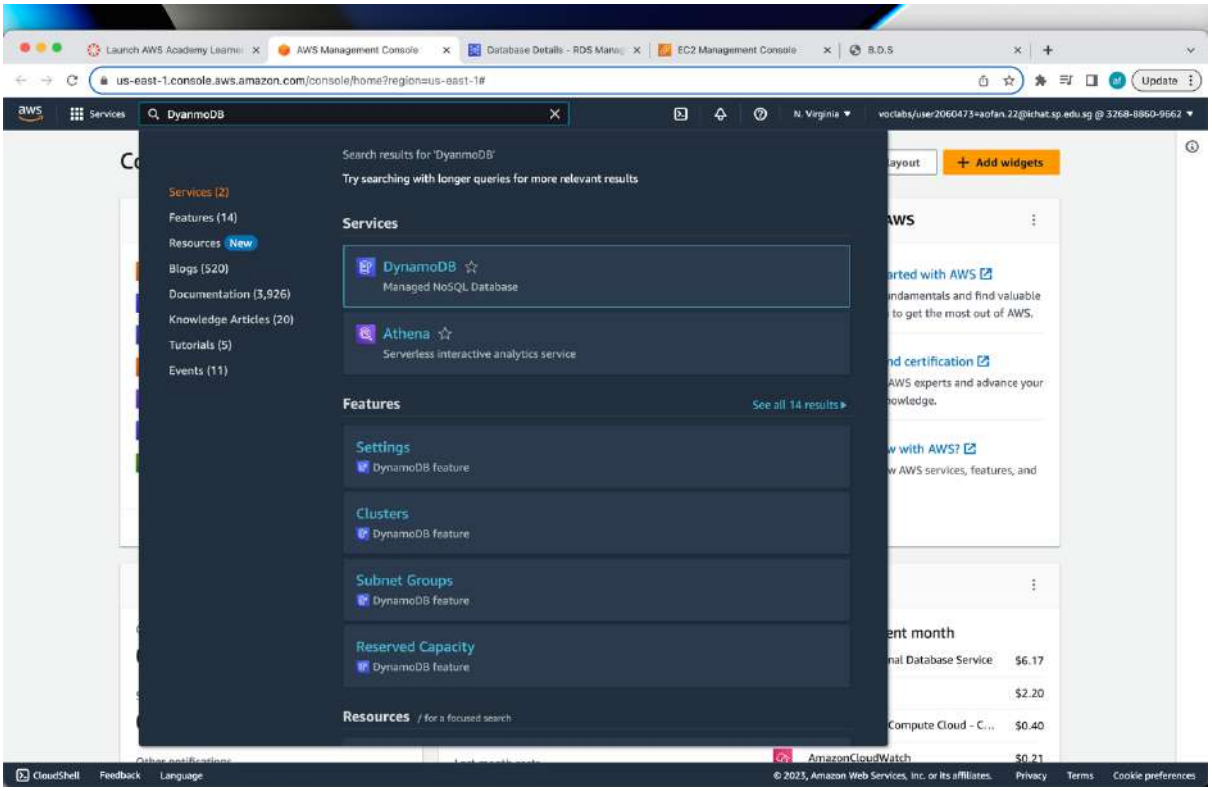




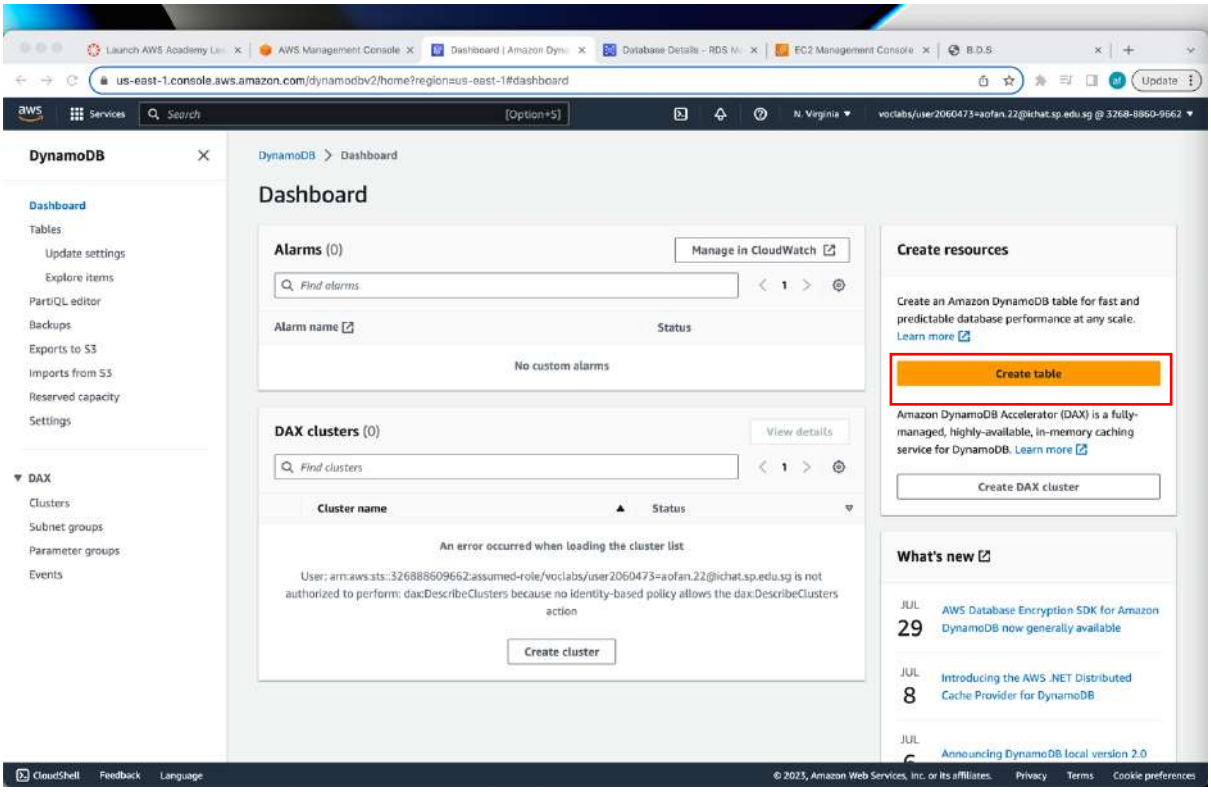
- The file exported should look like this in TextEdit.



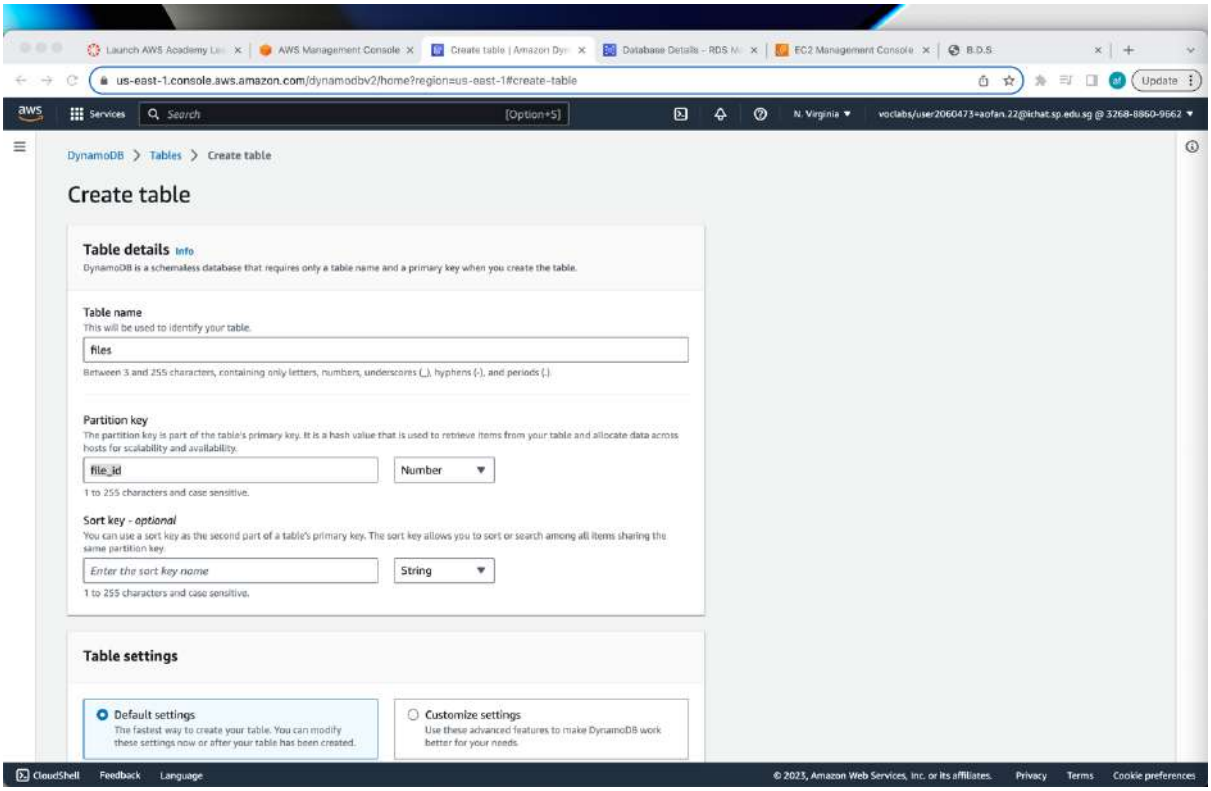
- Navigate to DynamoDB via AWS Management Console.



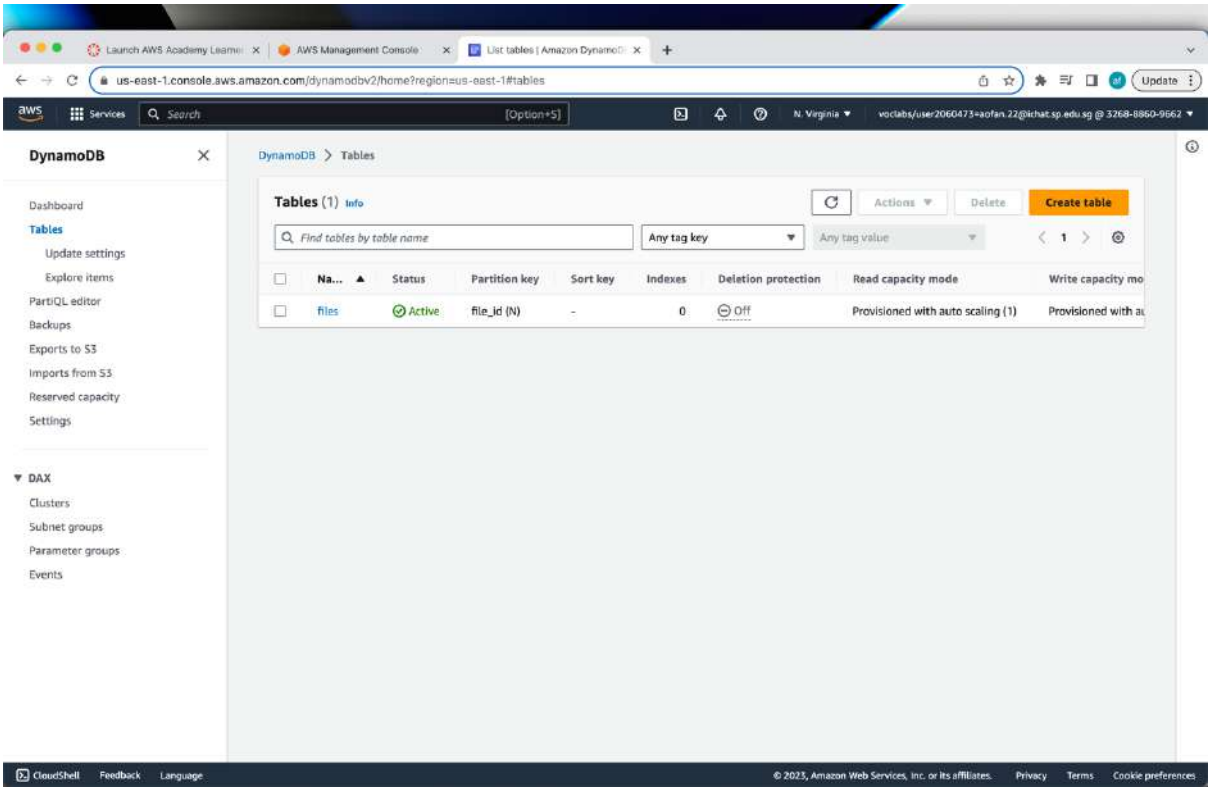
- Click Create table.



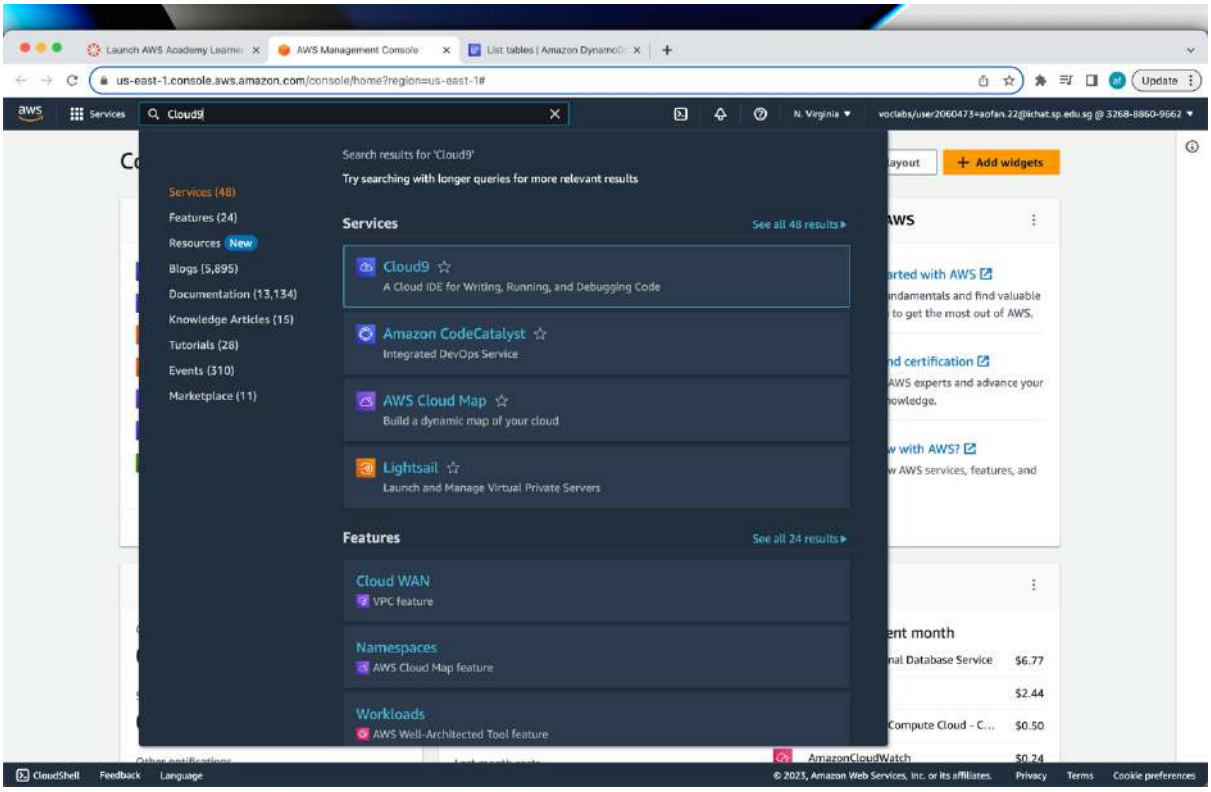
- Set Table name to be files, partition key name to be file_id, select Number from the dropdown list and scroll to the bottom of the page and click on Create table.



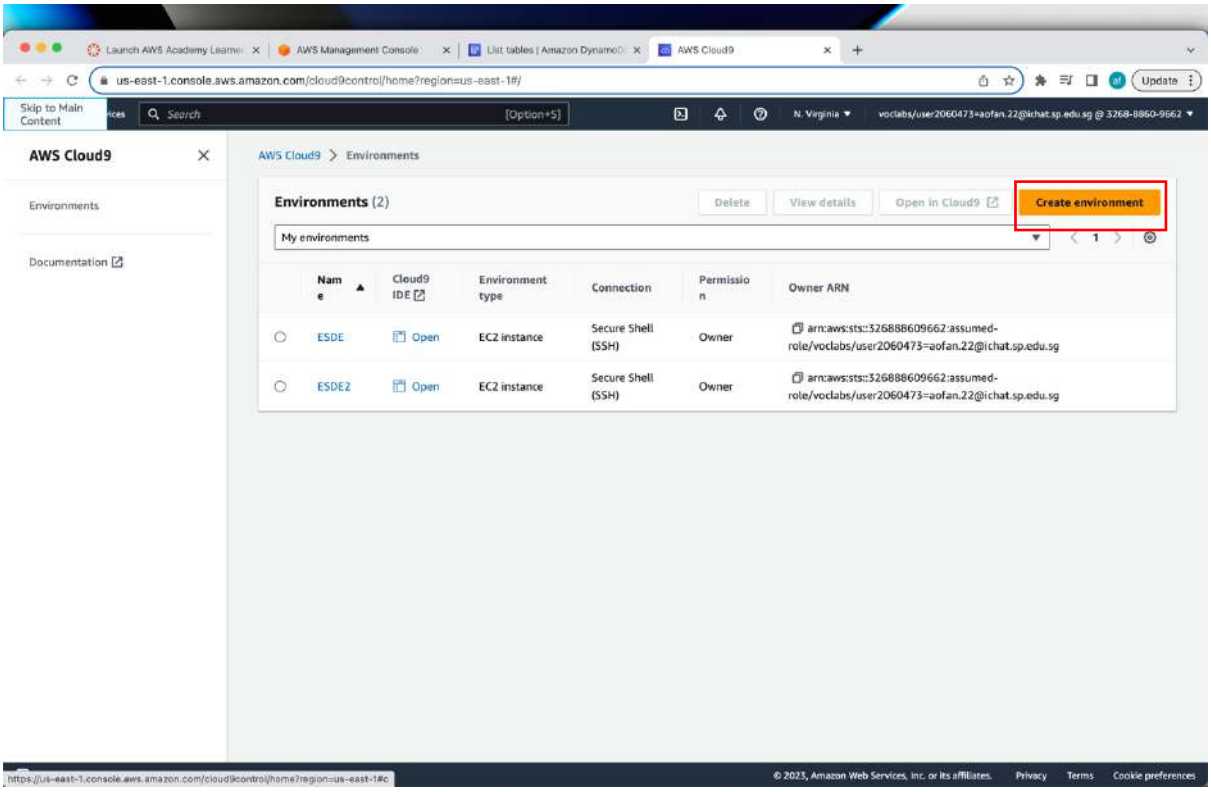
- Head to DynamoDB Tables tab to check the newly created table.



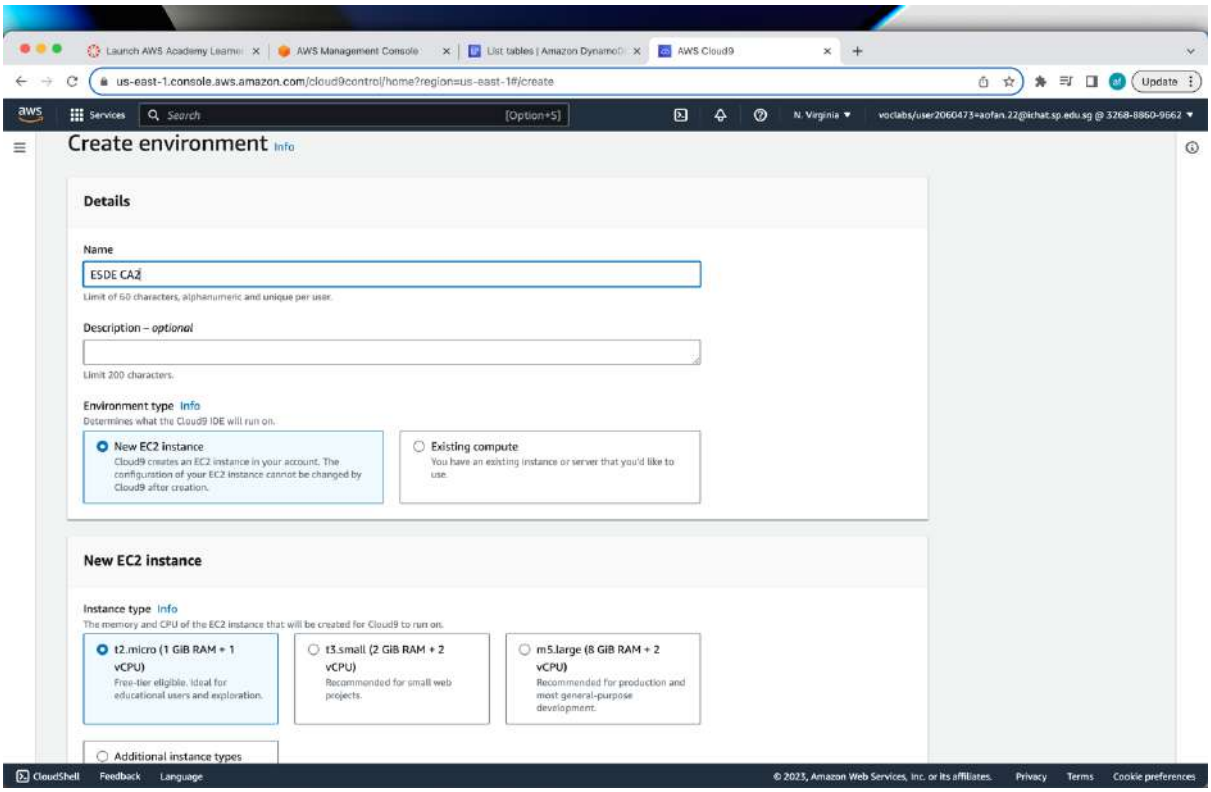
- Navigate to Cloud9



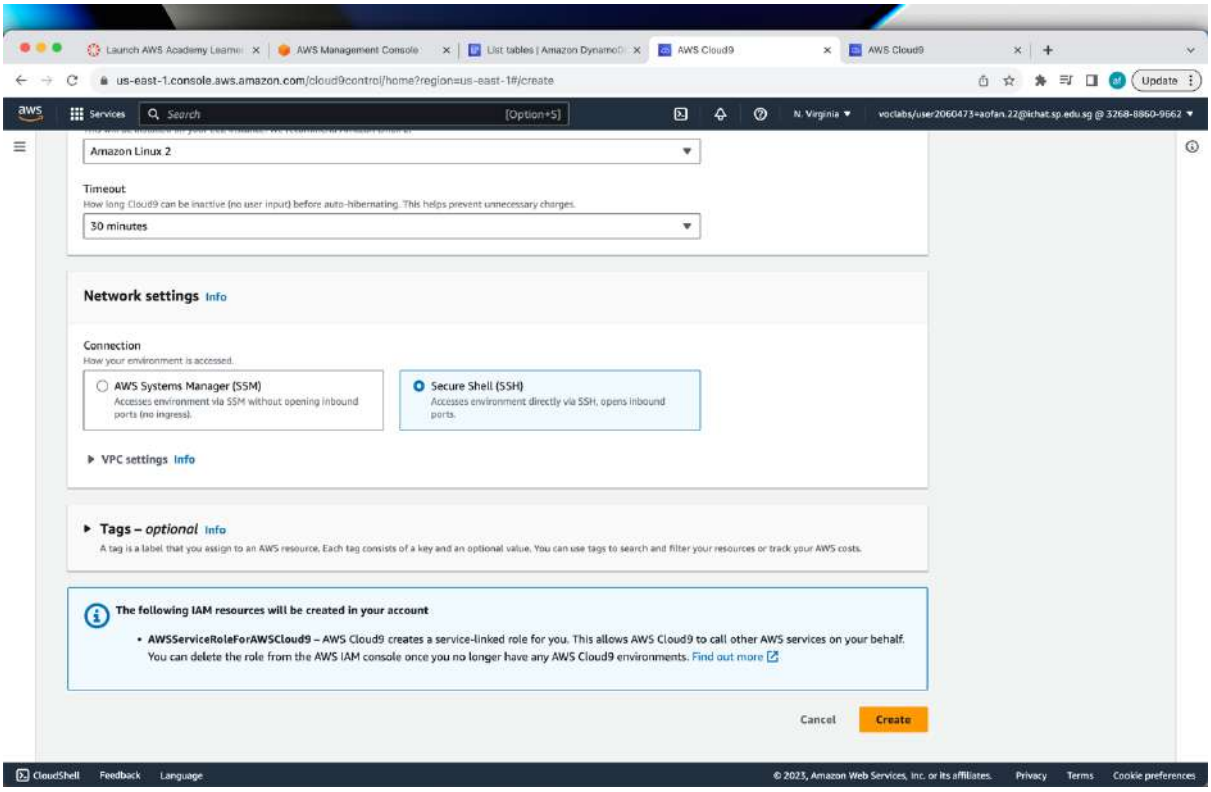
- Click on the Create environment button



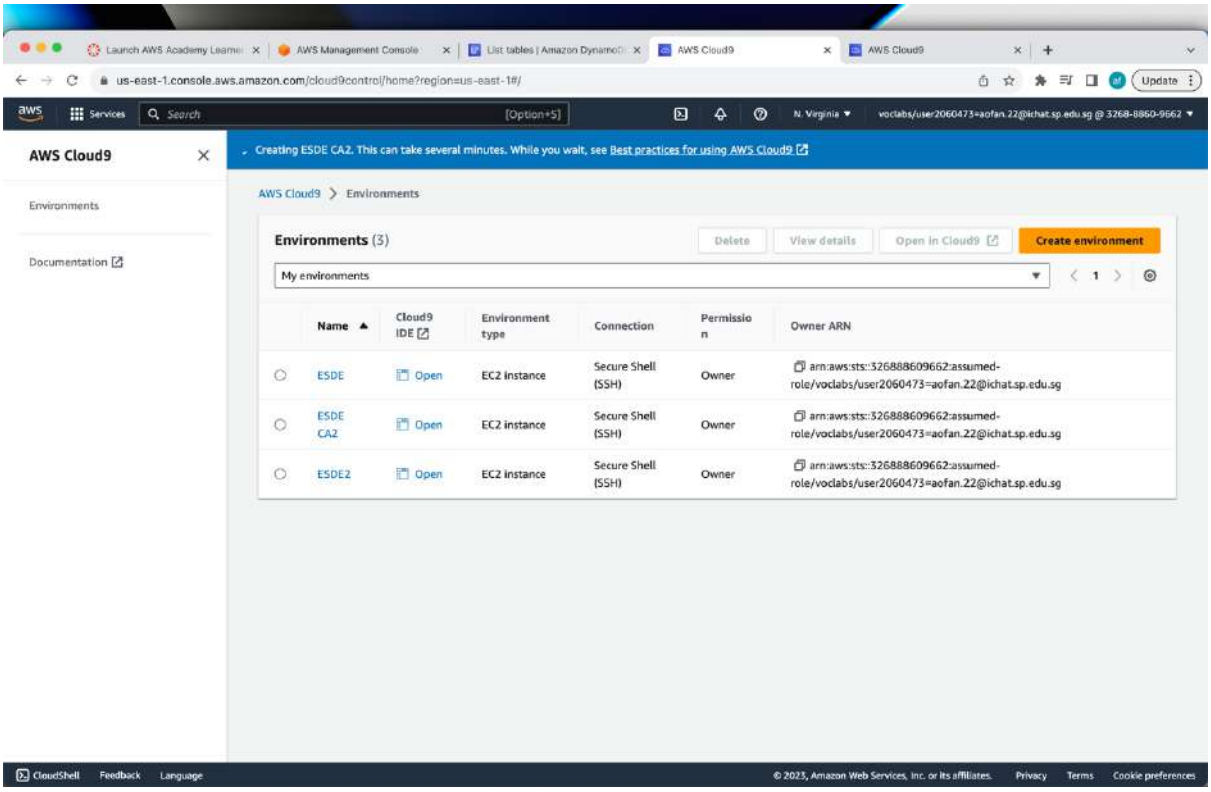
- Set Name to be ESDE CA2, select New EC2 instance under Environment type and choose t2.micro(1 GiB RAM + 1 vCPU) under Instance type.

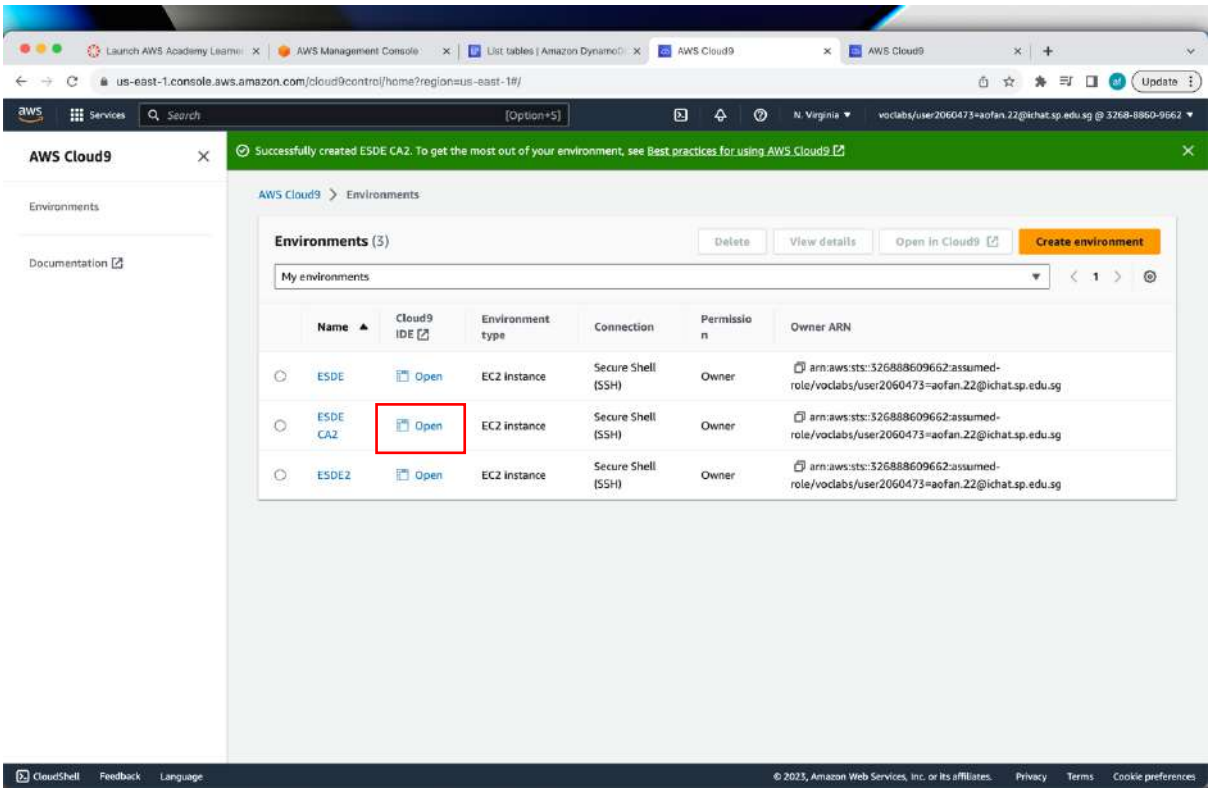


- Select Secure Shell (SSH) under Network settings, leave everything else as default and click on the Create button.

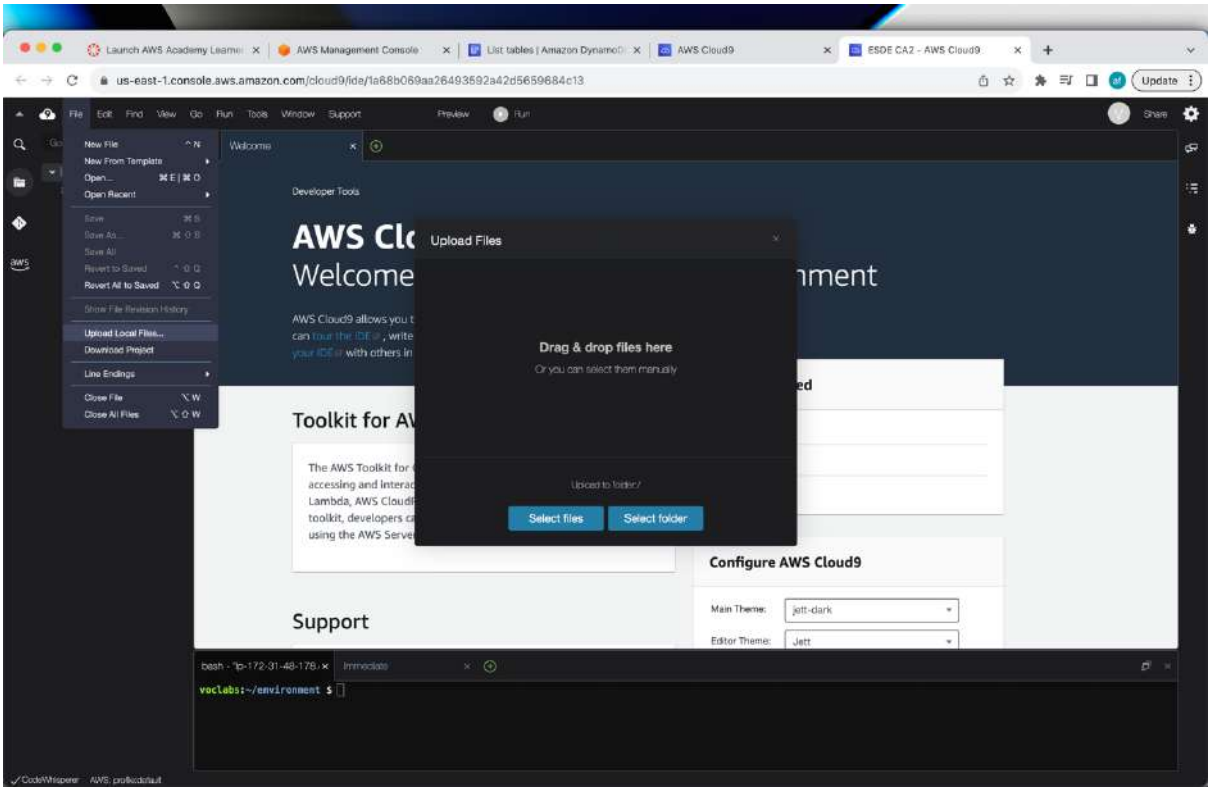


- You will be redirected back the AWS Cloud9 Environment, wait for the environment to set up then click Open.

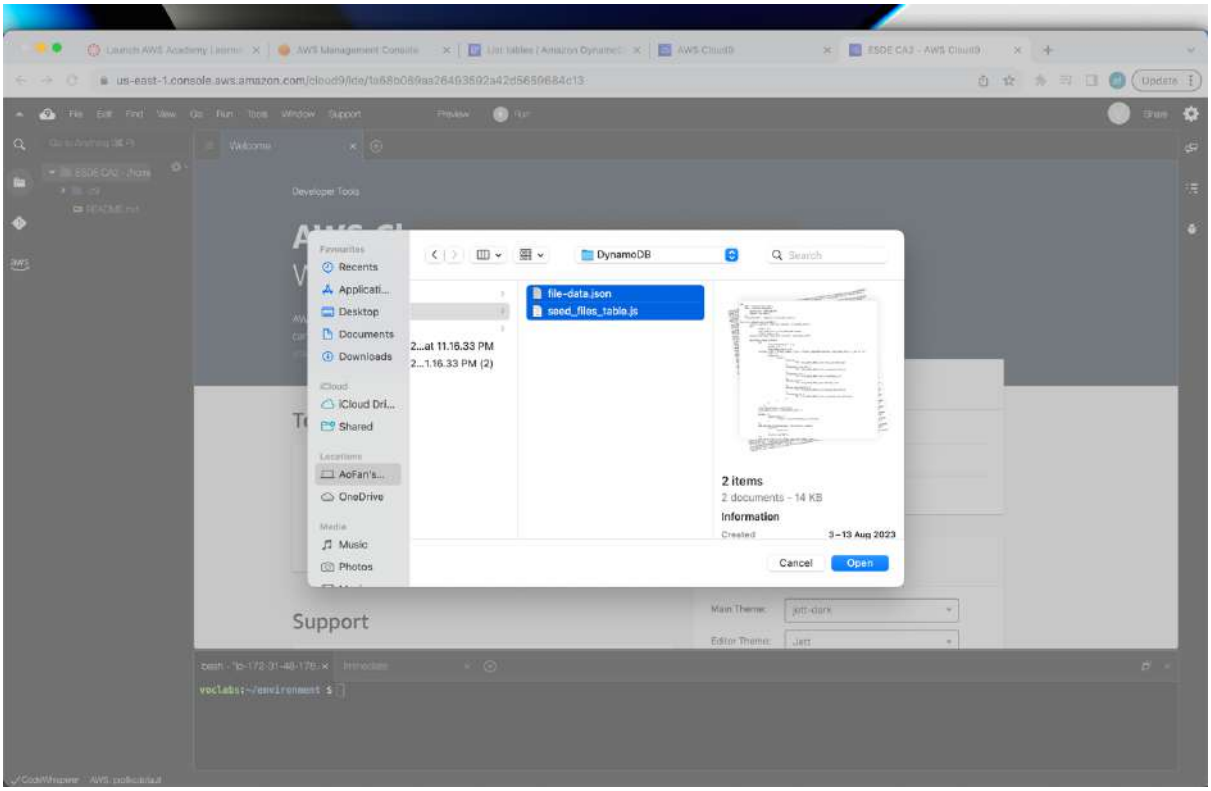




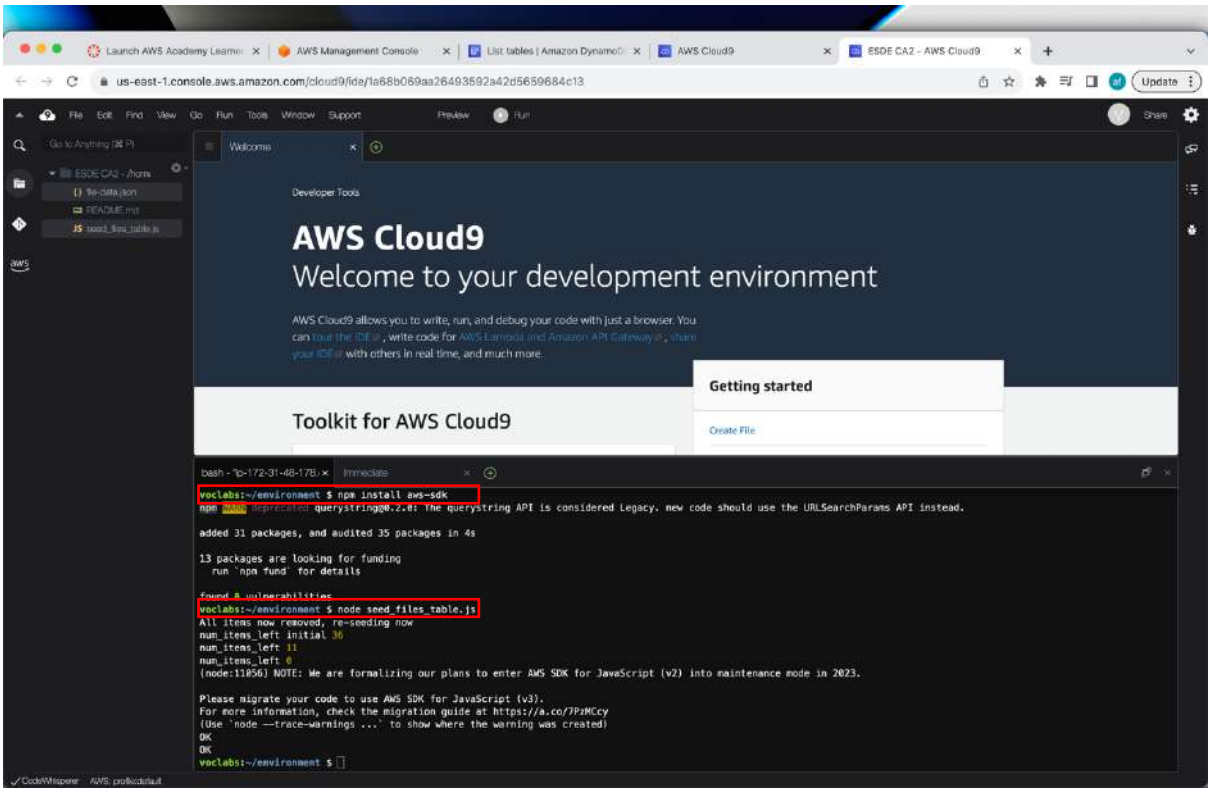
- Select File then click on Upload Local Files.... Upload Files prompt will appear, click on Select files.



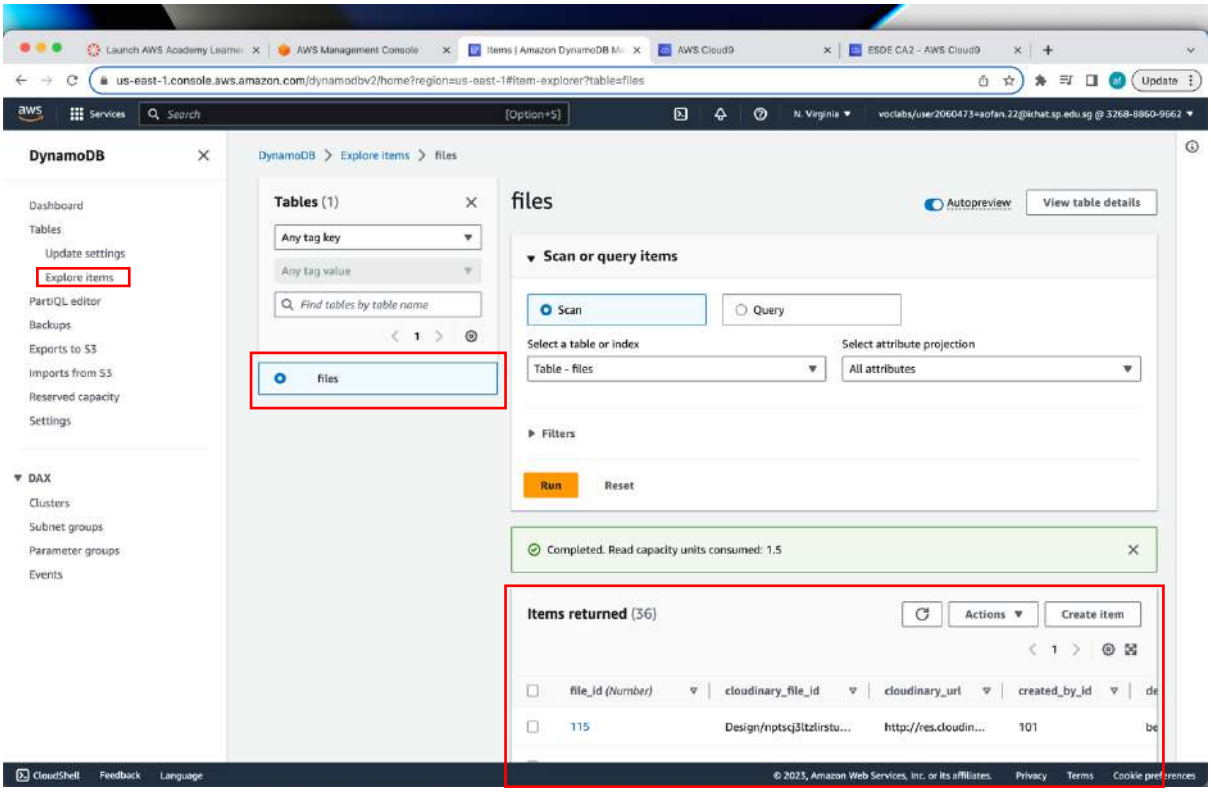
- Select file-data.json and seed_files_tables.js and click open.



- The two files will be uploaded in the environment. In terminal, run npm install aws-sdk followed by node seed_files_table.js.

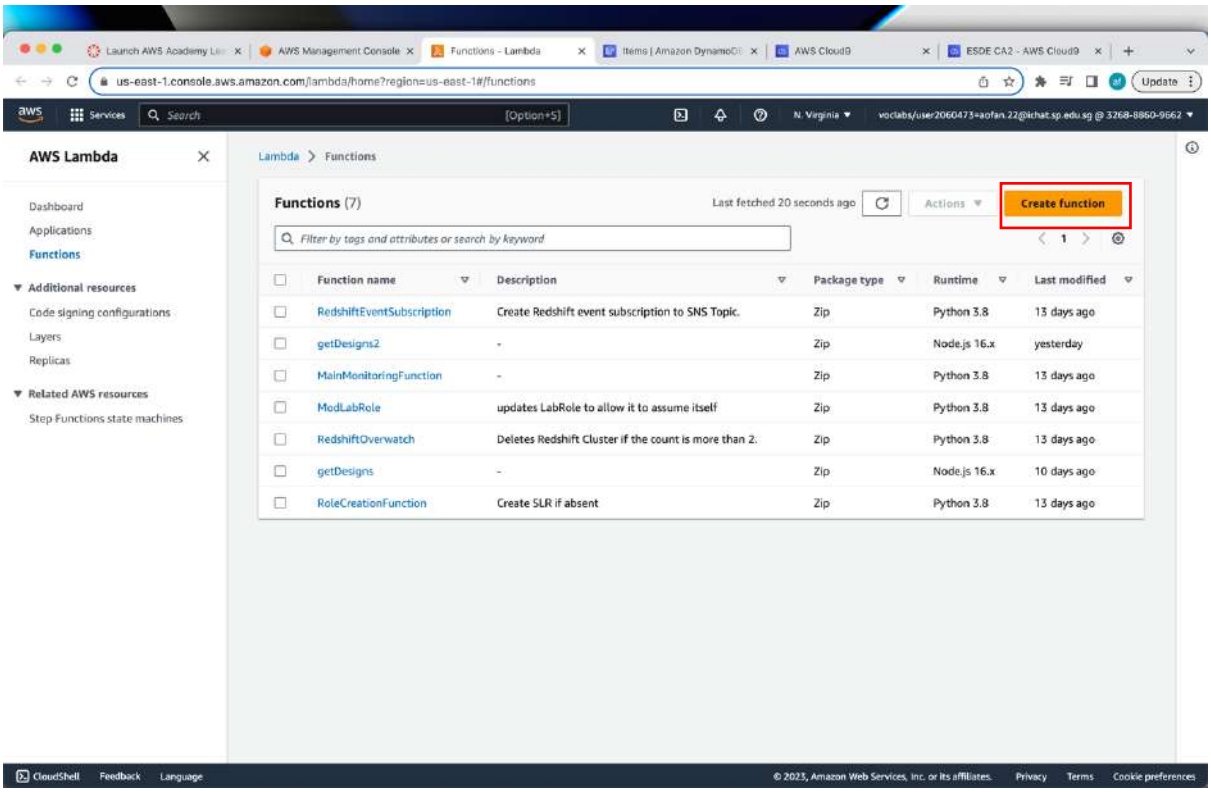


- Head to DynamoDB, select Explore items followed by the files table to confirm that the table has been populated.

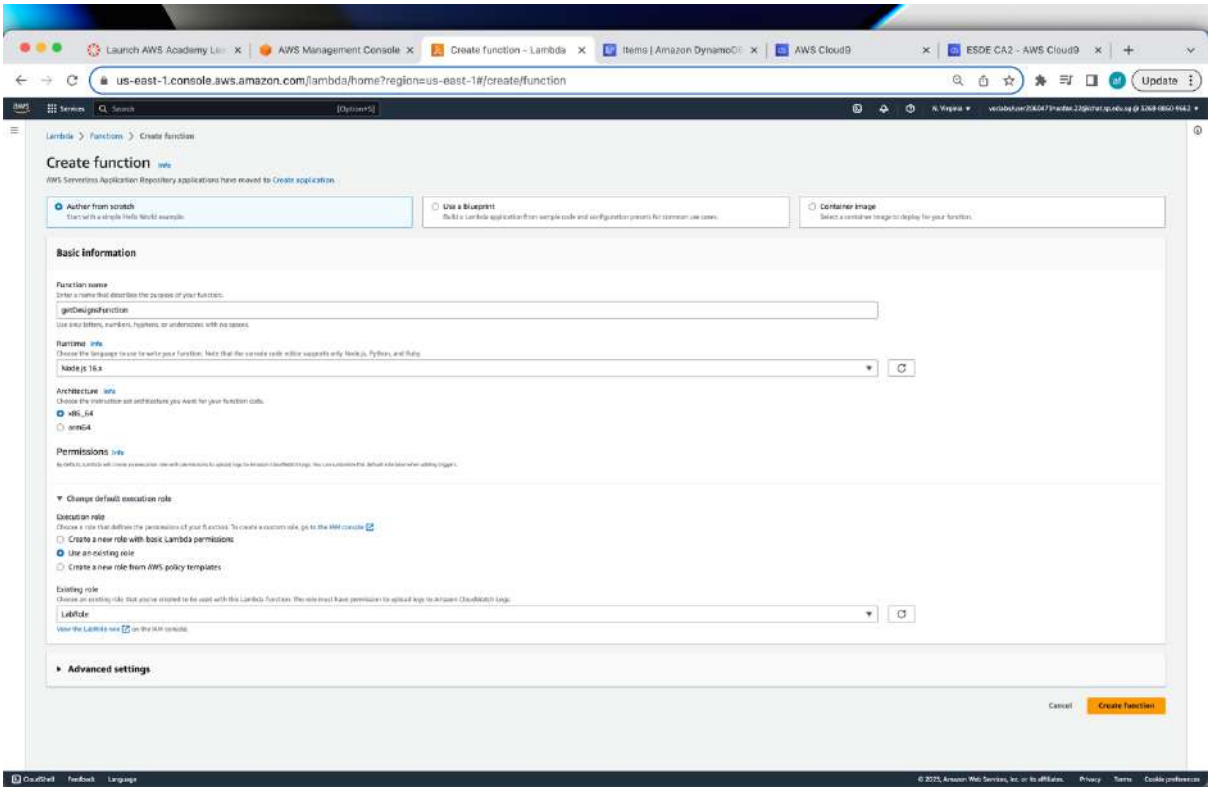

DynamoDB
Dashboard
Tables
Update settings
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
DynamoDB > Explore items > files
Tables (1)
Any tag key
Any tag value
Find tables by table name
files
files
Autopreview
View table details
Scan or query items
Scan
Query
Select a table or index
Table - files
Select attribute projection
All attributes
Filters
Run
Reset
Completed. Read capacity units consumed: 1.5
Items returned (36)
Actions
Create item
file_id (Number)
cloudinary_file_id
cloudinary_url
created_by_id
115
Design/nptscj3ltzlirstu...
http://res.cloudin...
101
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

7. Creating and testing ‘getDesginsFunction’ Lambda function

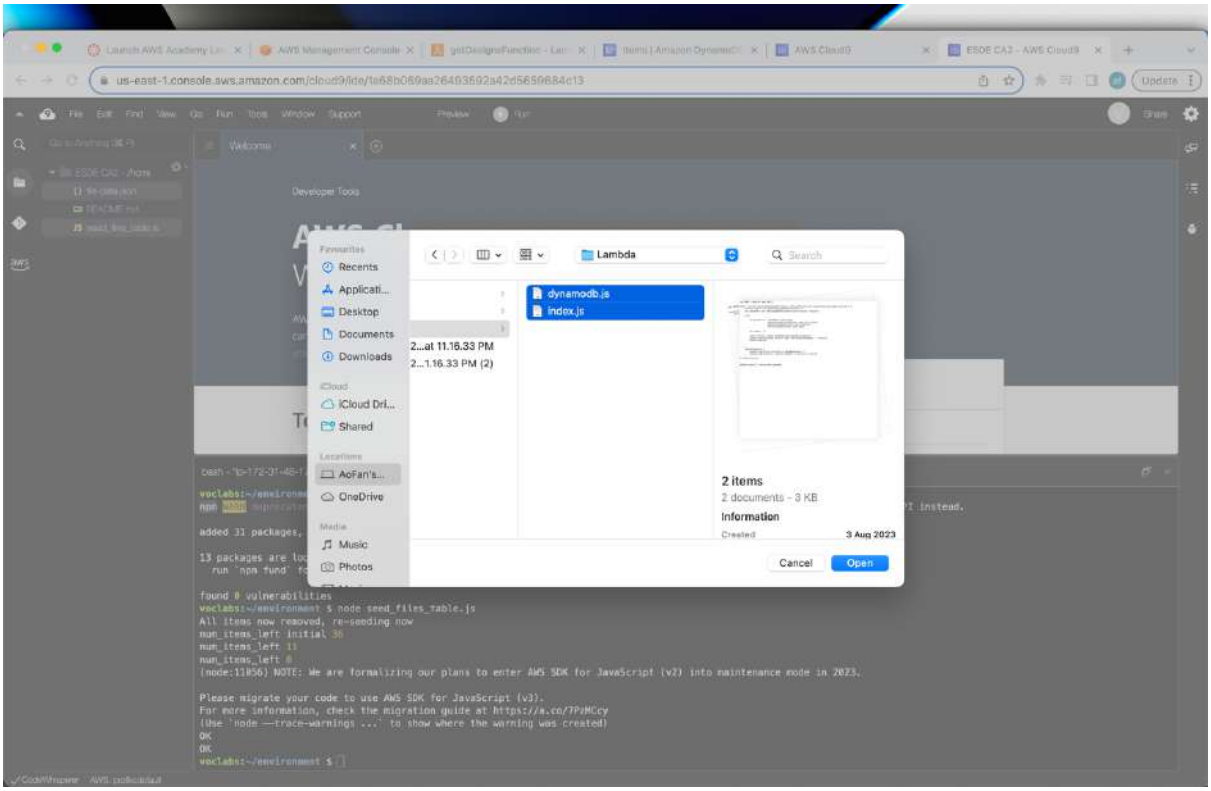
- Navigate to Lambda service console and click Create function.



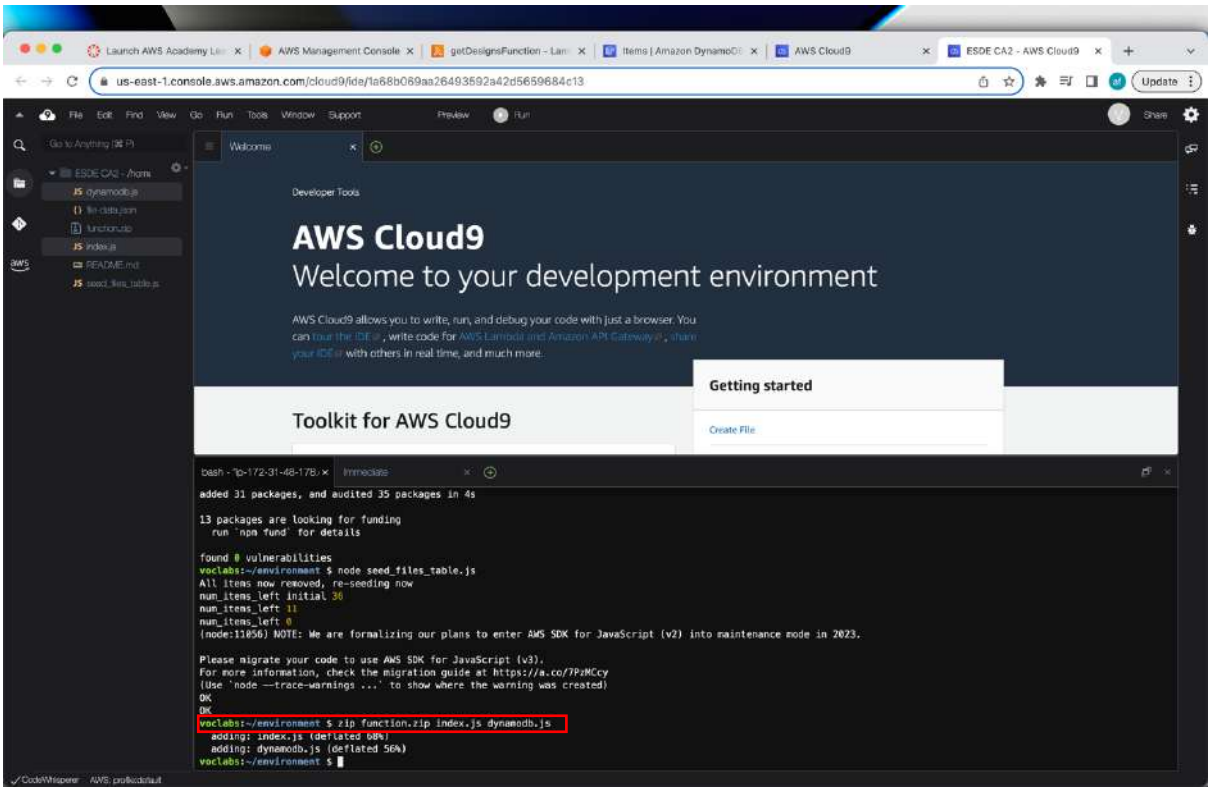
- For the creation of the function, follow the illustration below.



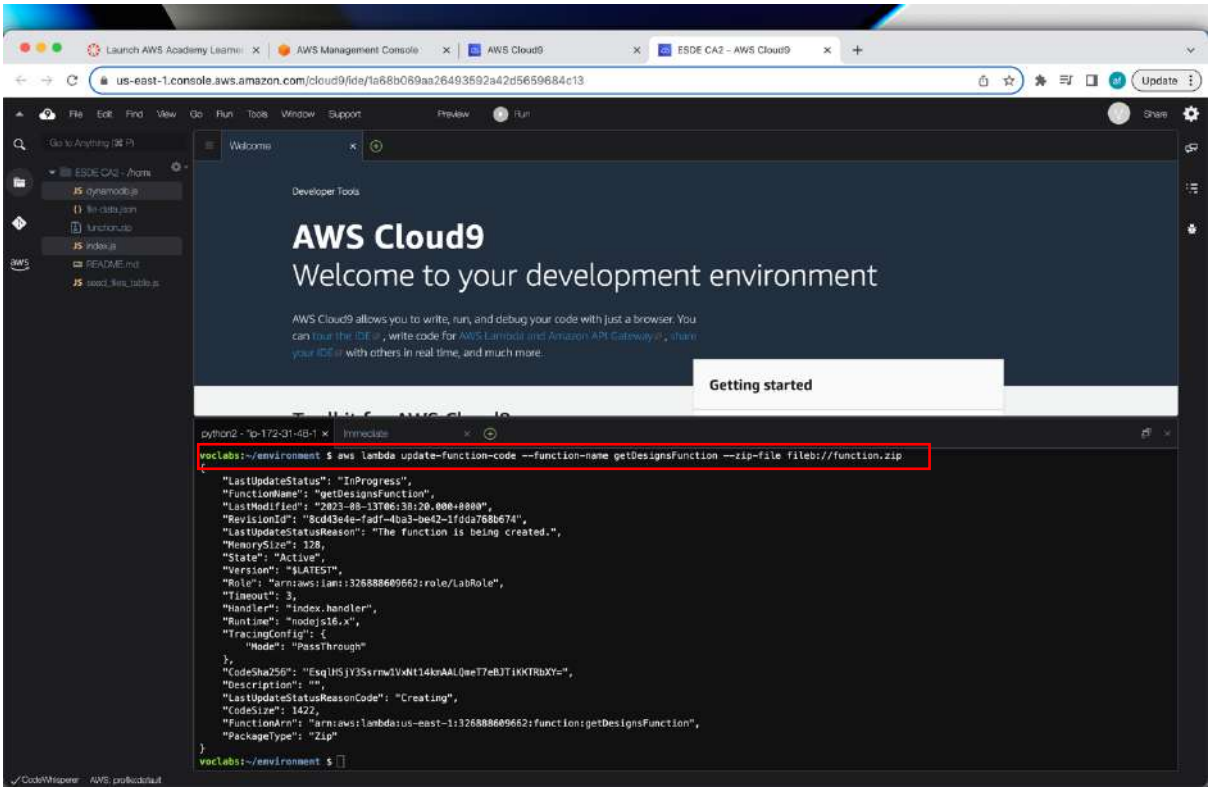
- After creating the function, go to Cloud9 and open IDE. Upload dynamodb.js and index.js files.



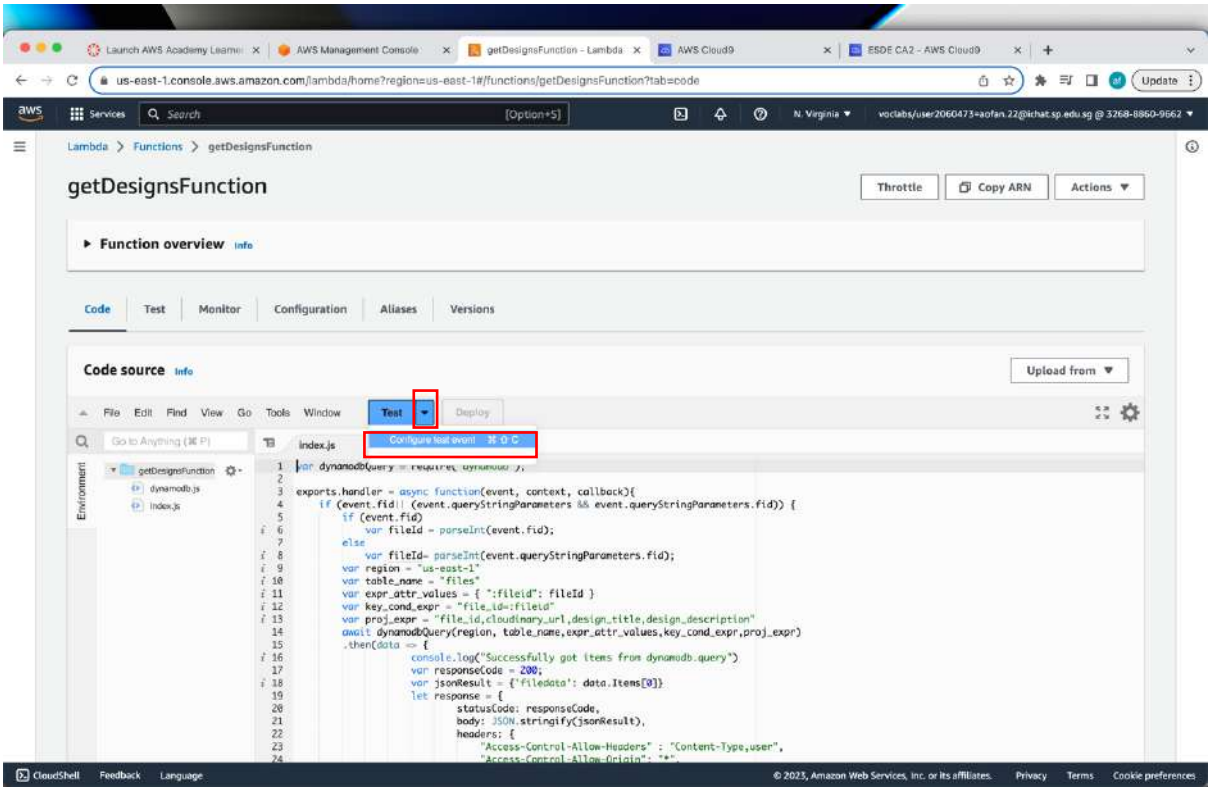
- Run zip function.zip index.js dynamodb.js.



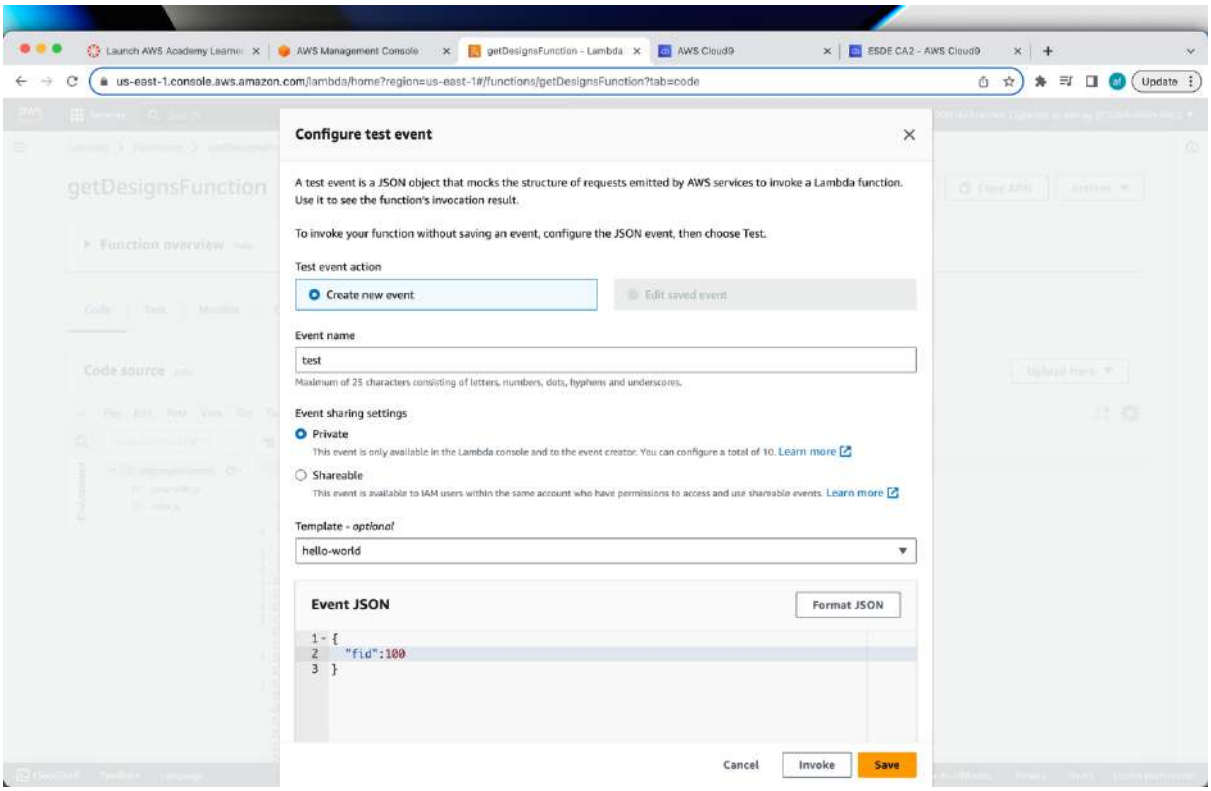
- Next, run aws lambda update-function-code --function-name getDesignsFunction --zip-file fileb://function.zip.



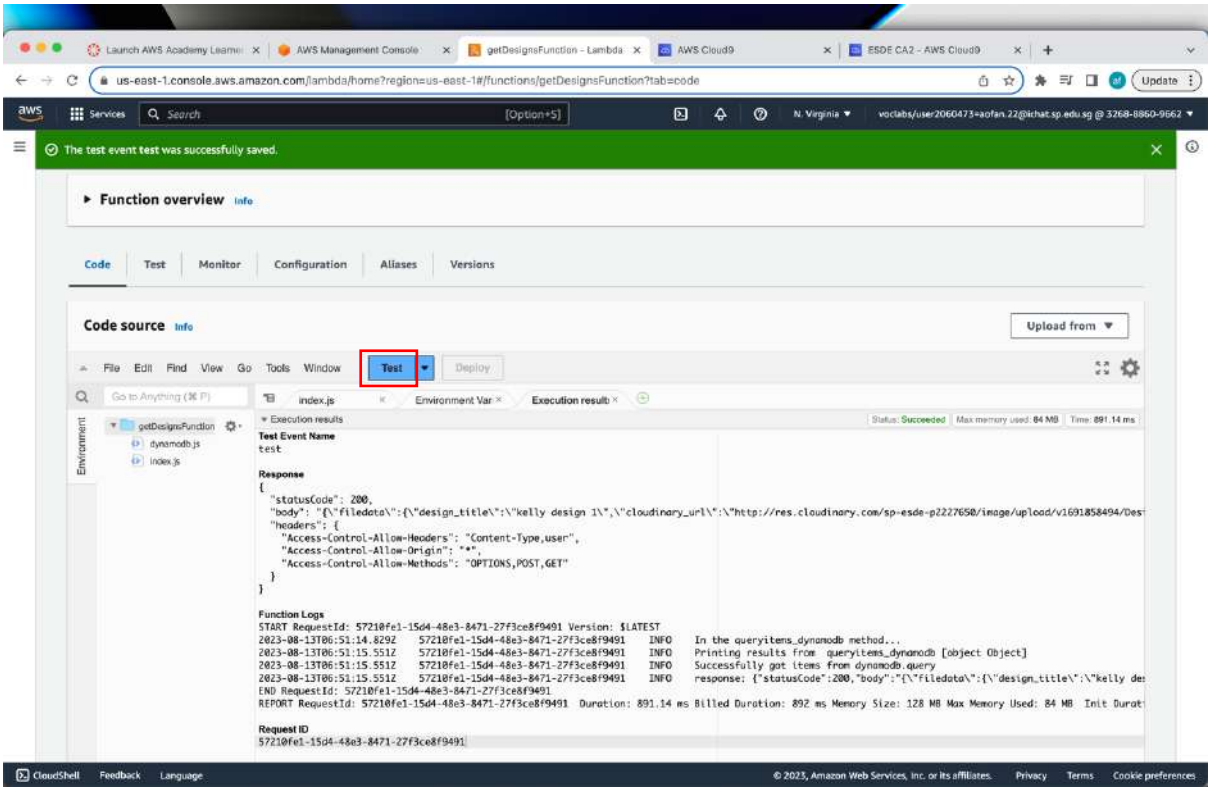
- Go to AWS Lambda > Functions > getDesignsFunction. (the files in the zip file should appear here) Click on the dropdown arrow next to Test and choose Configure test event.



- Set up the test event as shown in the image below.

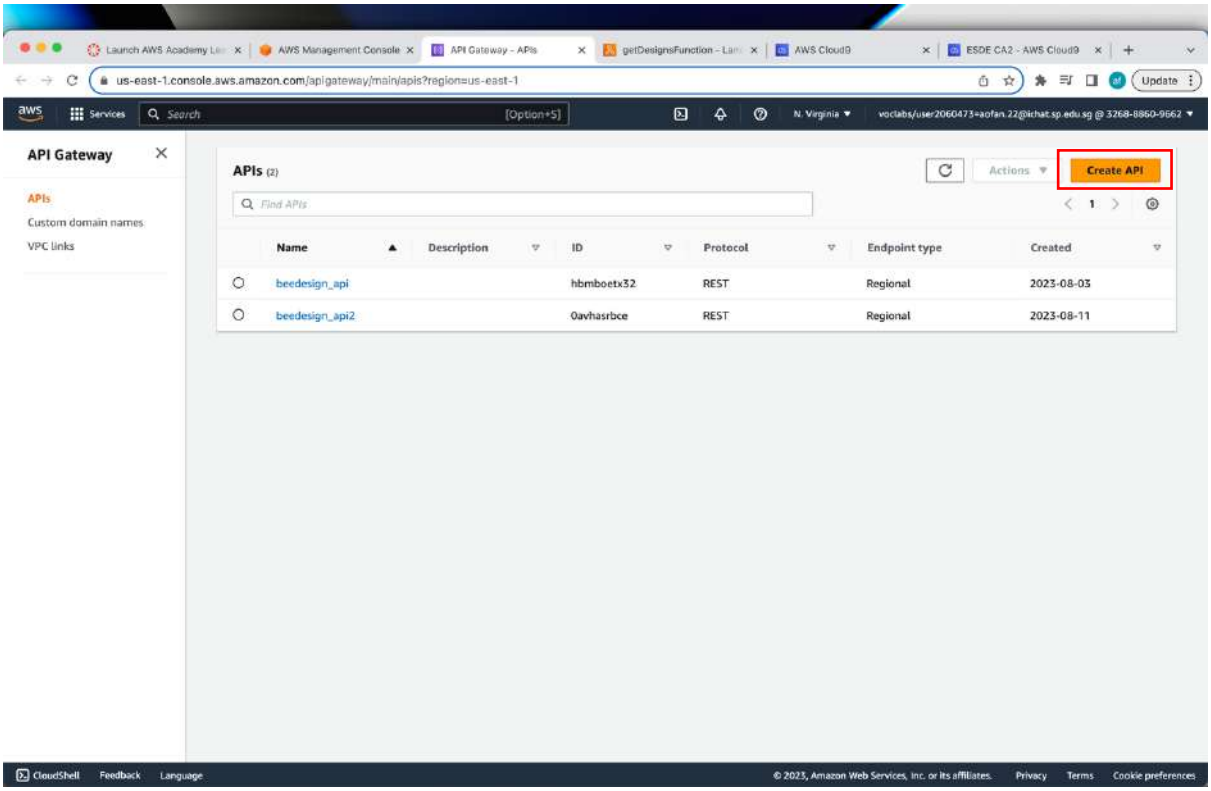


- Hit the Save button and click on Test. If the Lambda function is configured correctly, response similar to the illustration below will be displayed.

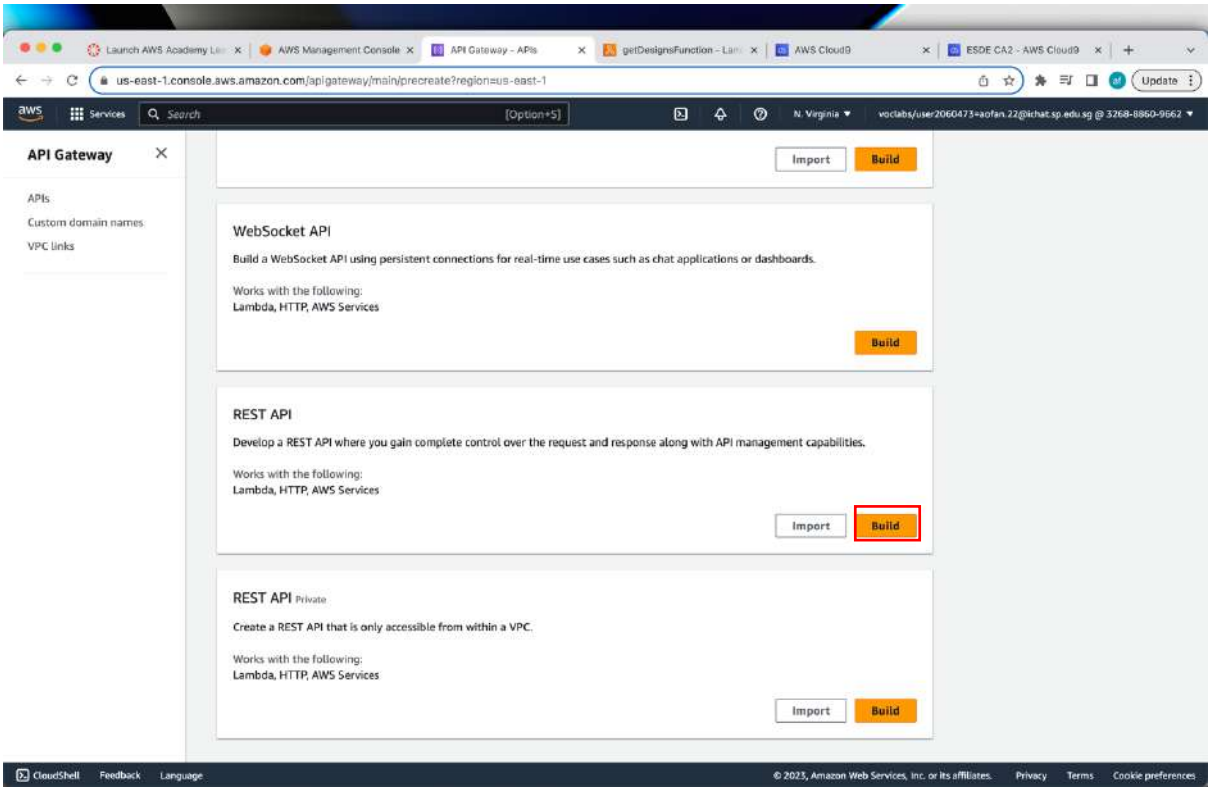

Launch AWS Academy Learner
AWS Management Console
getDesignsFunction - Lambda
AWS Cloud9
ESDE CA2 - AWS Cloud9
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/getDesignsFunction?tab=code
Update
Services
Search
[Option+S]
N. Virginia
The test event test was successfully saved.
Function overview Info
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Upload from
File Edit Find View Go Tools Window
Test
Deploy
index.js
Environment Var
Execution result
Execution results
Status: Succeeded
Max memory used: 84 MB
Time: 891.14 ms
Environment
getDesignsFunction
dynamodb.js
index.js
Test Event Name
test
Response
{
"statusCode": 200,
"body": "{\"filedata\":{\"design_title\":\"kelly design 1\",\"cloudinary_url\":\"http://res.cloudinary.com/sp-esde-p2227650/image/upload/v1691858494/Des
"headers": {
"Access-Control-Allow-Headers": "Content-Type,user",
"Access-Control-Allow-Origin": "*",
"Access-Control-Allow-Methods": "OPTIONS,POST,GET"
}
}
Function Logs
START RequestId: 57210fe1-15d4-48e3-8471-27f3ce8f9491 Version: $LATEST
2023-08-13T06:51:14.829Z 57210fe1-15d4-48e3-8471-27f3ce8f9491 INFO In the queryitems_dynamodb method...
2023-08-13T06:51:15.551Z 57210fe1-15d4-48e3-8471-27f3ce8f9491 INFO Printing results from queryitems_dynamodb [object Object]
2023-08-13T06:51:15.551Z 57210fe1-15d4-48e3-8471-27f3ce8f9491 INFO Successfully got items from dynamodb.query
2023-08-13T06:51:15.551Z 57210fe1-15d4-48e3-8471-27f3ce8f9491 INFO response: {"statusCode":200,"body":"{\"filedata\":{\"design_title\":\"kelly des
END RequestId: 57210fe1-15d4-48e3-8471-27f3ce8f9491
REPORT RequestId: 57210fe1-15d4-48e3-8471-27f3ce8f9491 Duration: 891.14 ms Billed Duration: 892 ms Memory Size: 128 MB Max Memory Used: 84 MB Init Durat
Request ID
57210fe1-15d4-48e3-8471-27f3ce8f9491
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

8. Setting up API endpoint for Lambda function using API Gateway

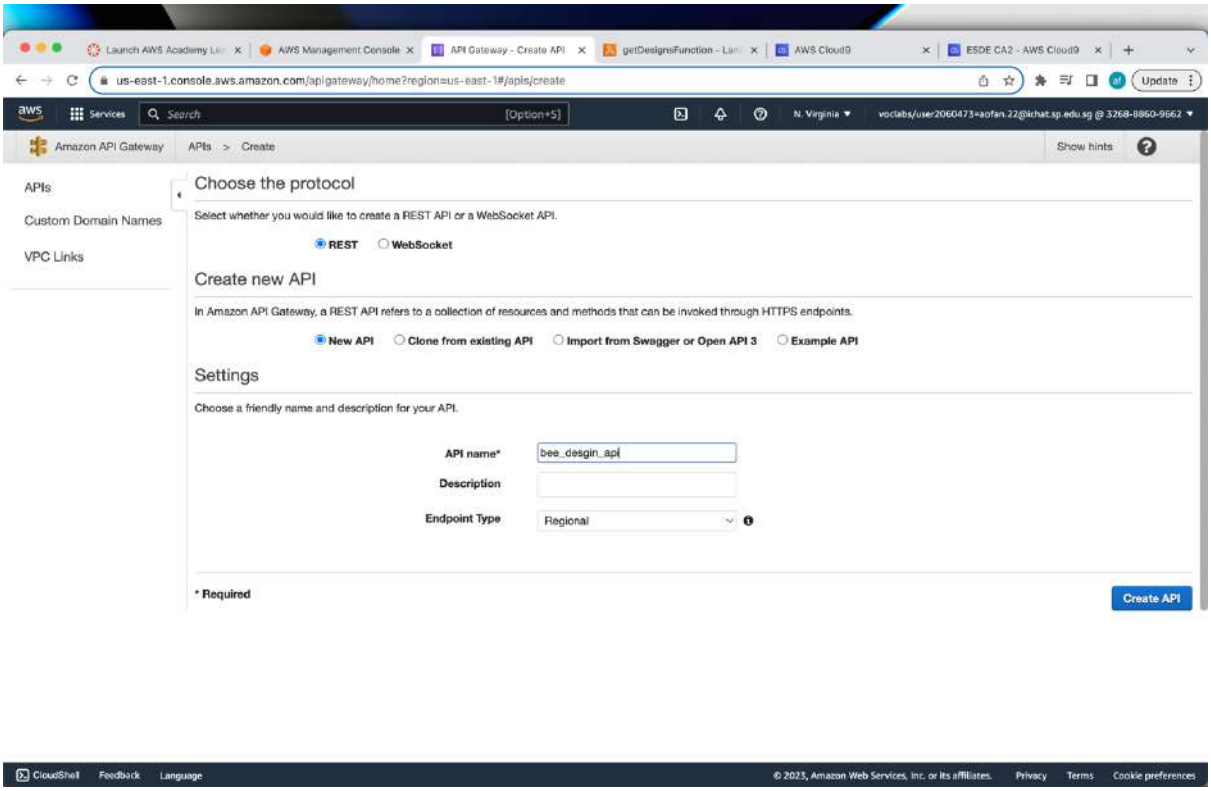
- Navigate to API Gateway and click Create API.



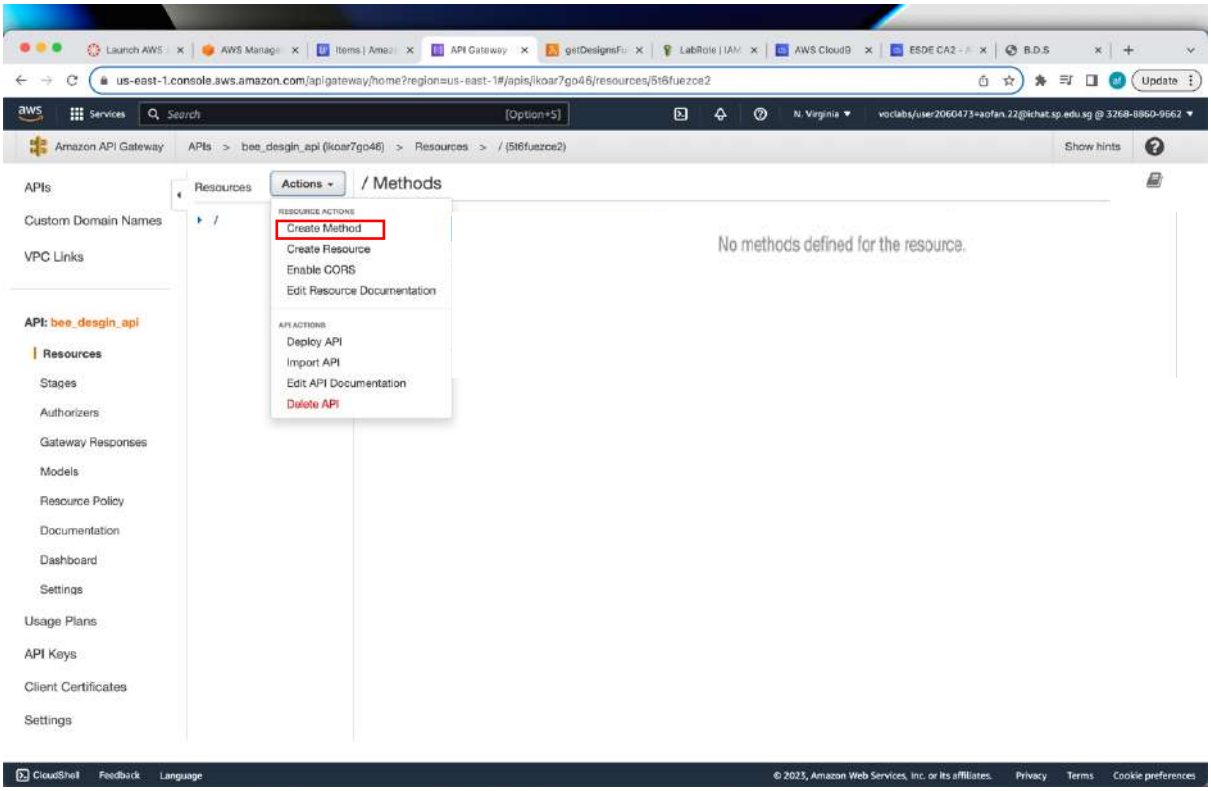
- Choose REST API and click build.



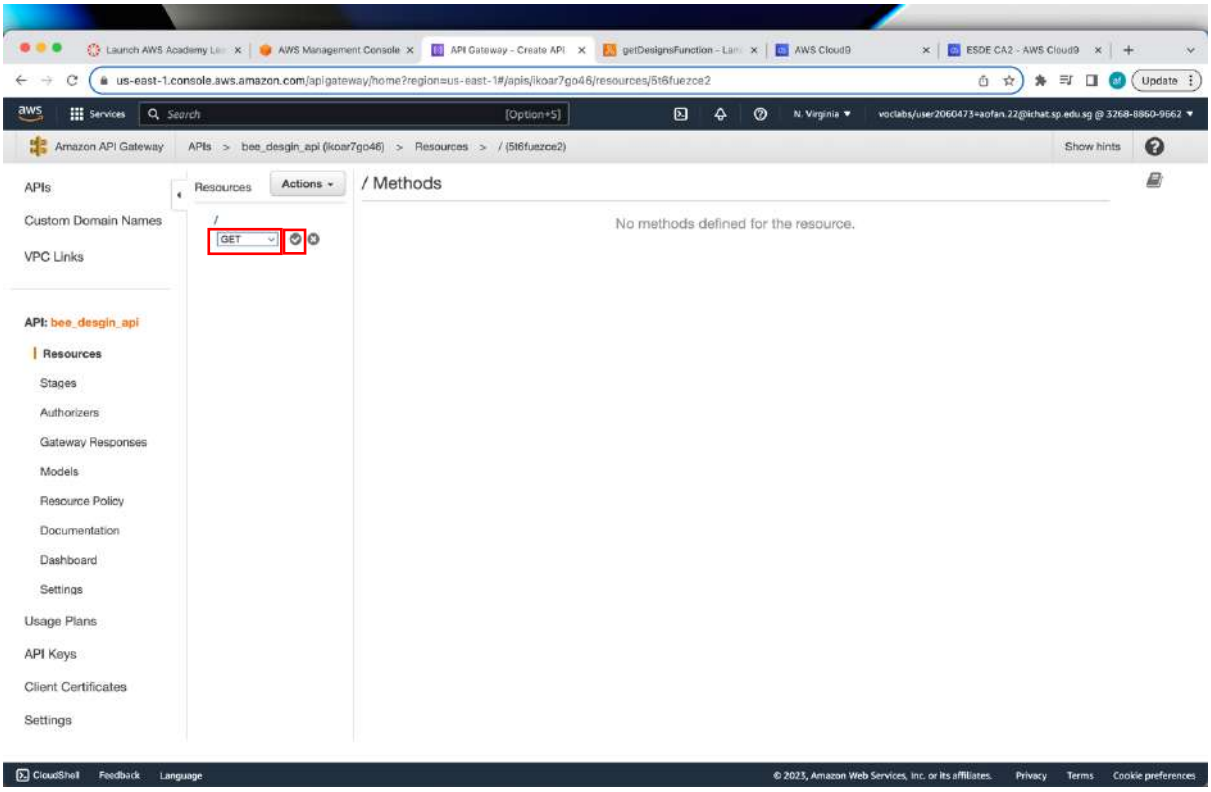
- Set up according to the illustration below and click Create API.



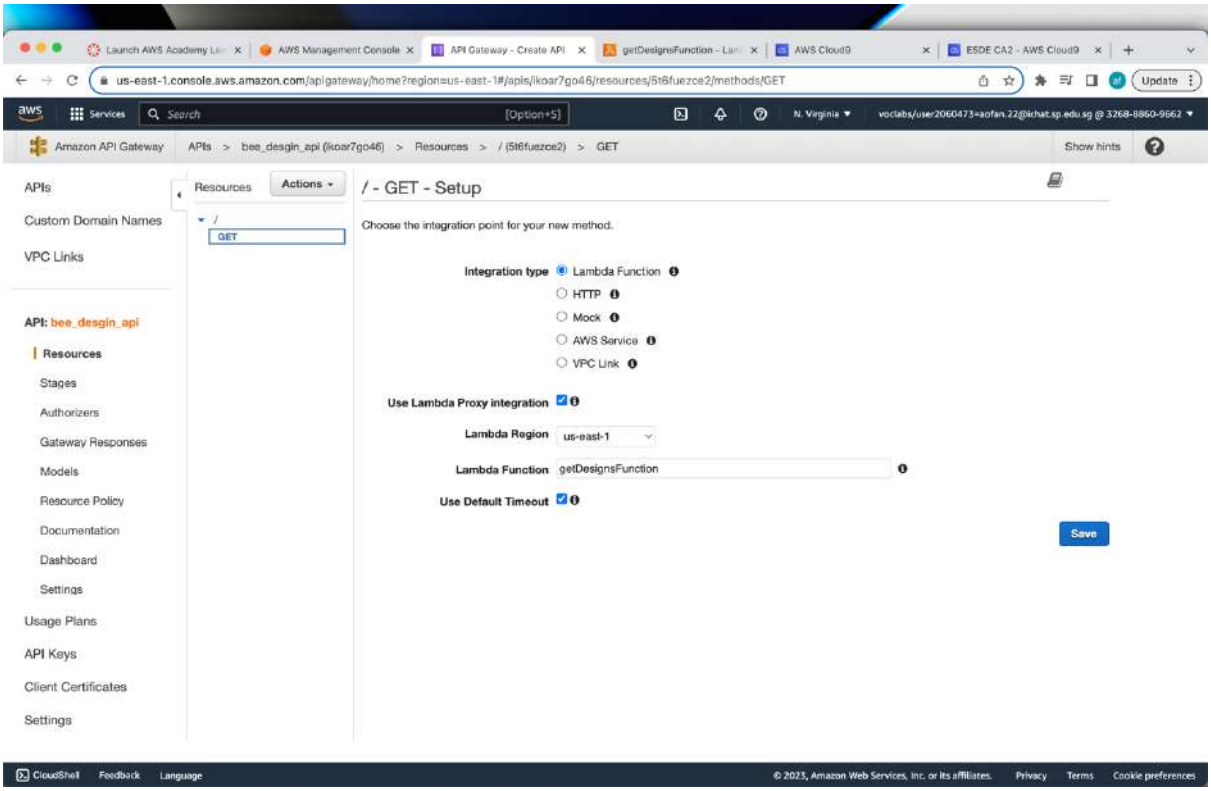
- After creating the API, click on Actions and choose Create Method.



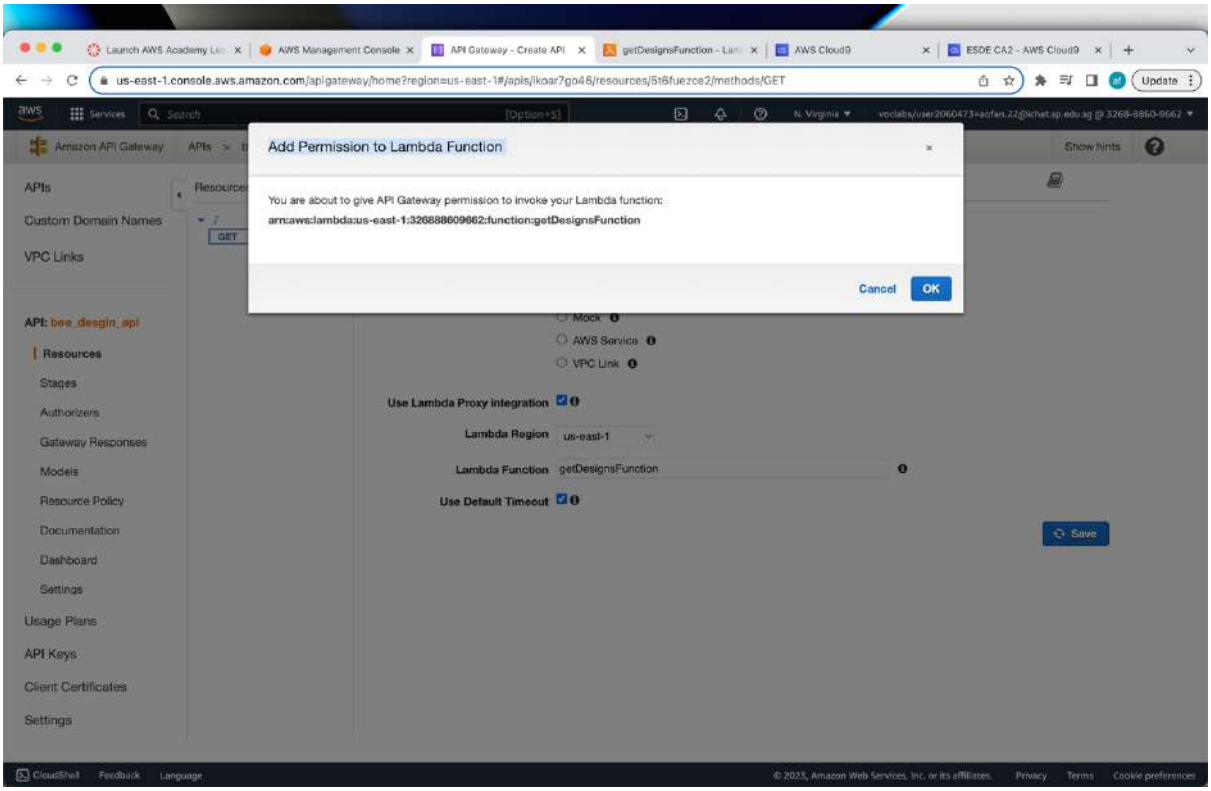
- Click on the dropdown box and select GET. Click on the tick to save.



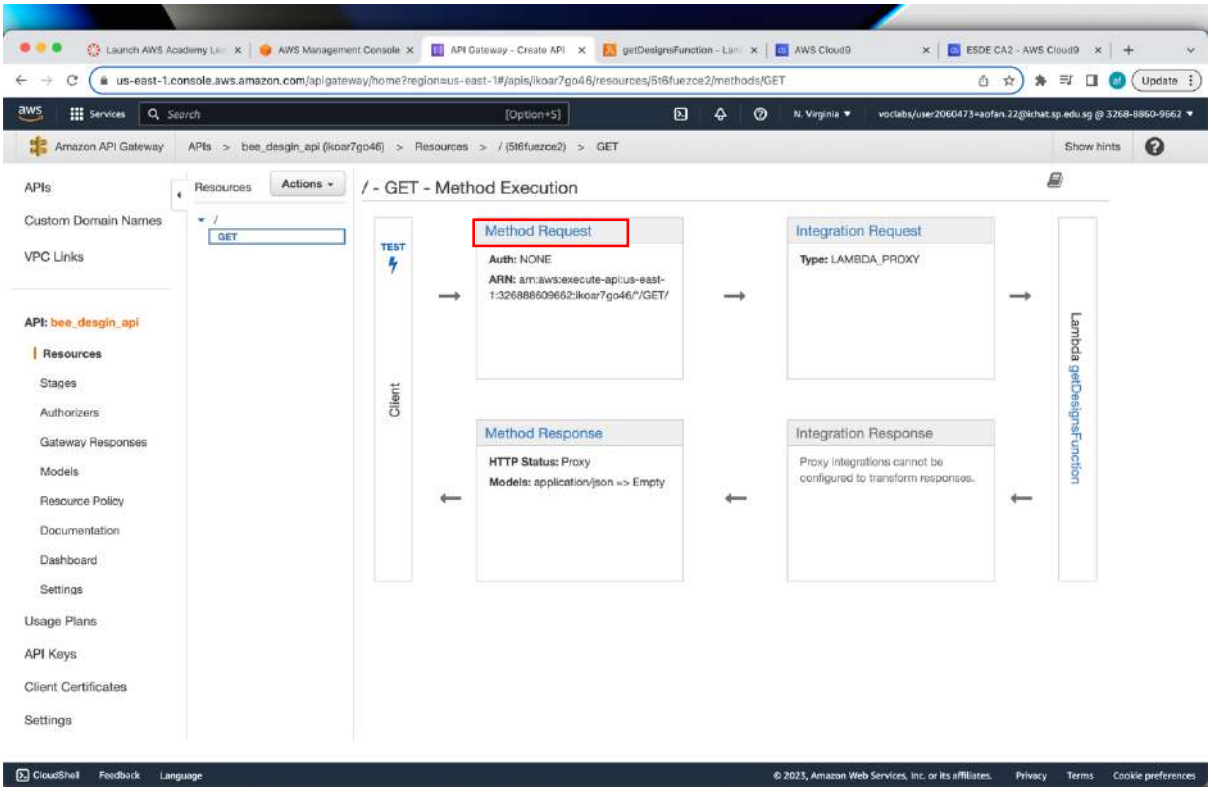
- Set up the method as shown below and click Save.



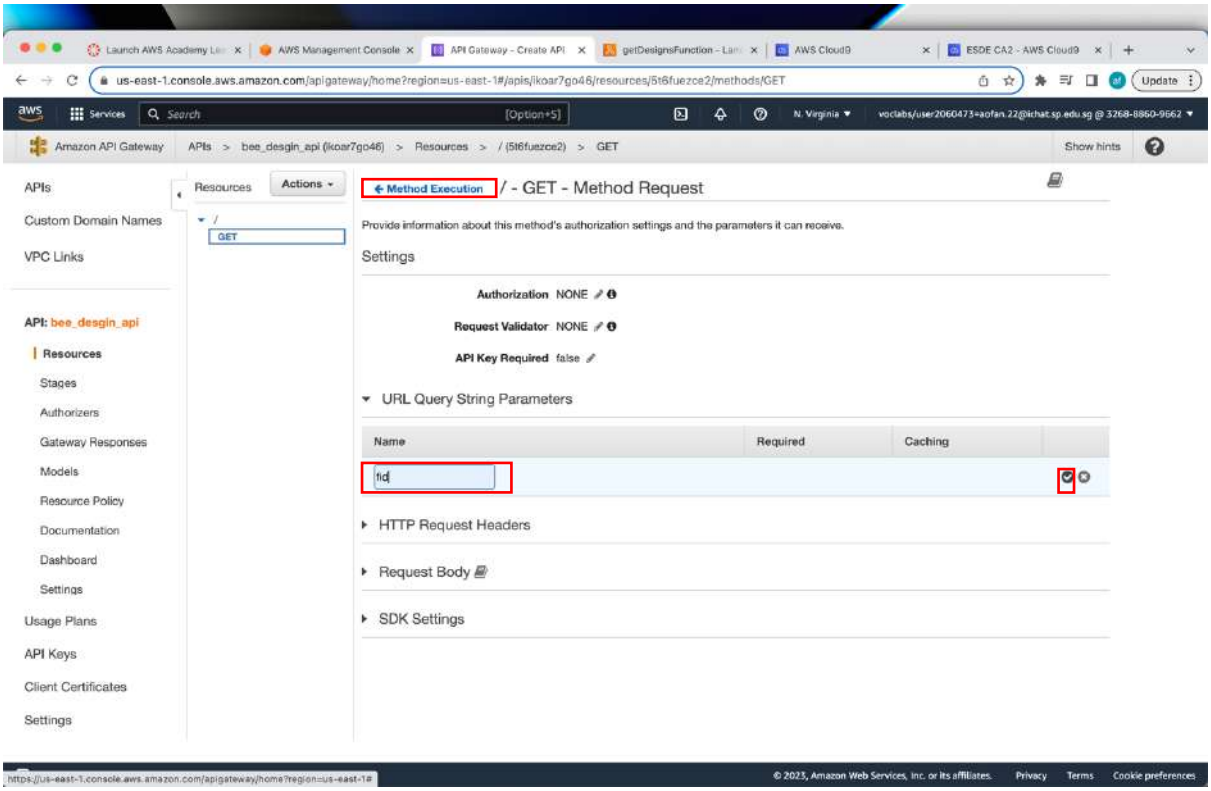
- Click OK when the Add Permission to Lambda Function prompt shows up.



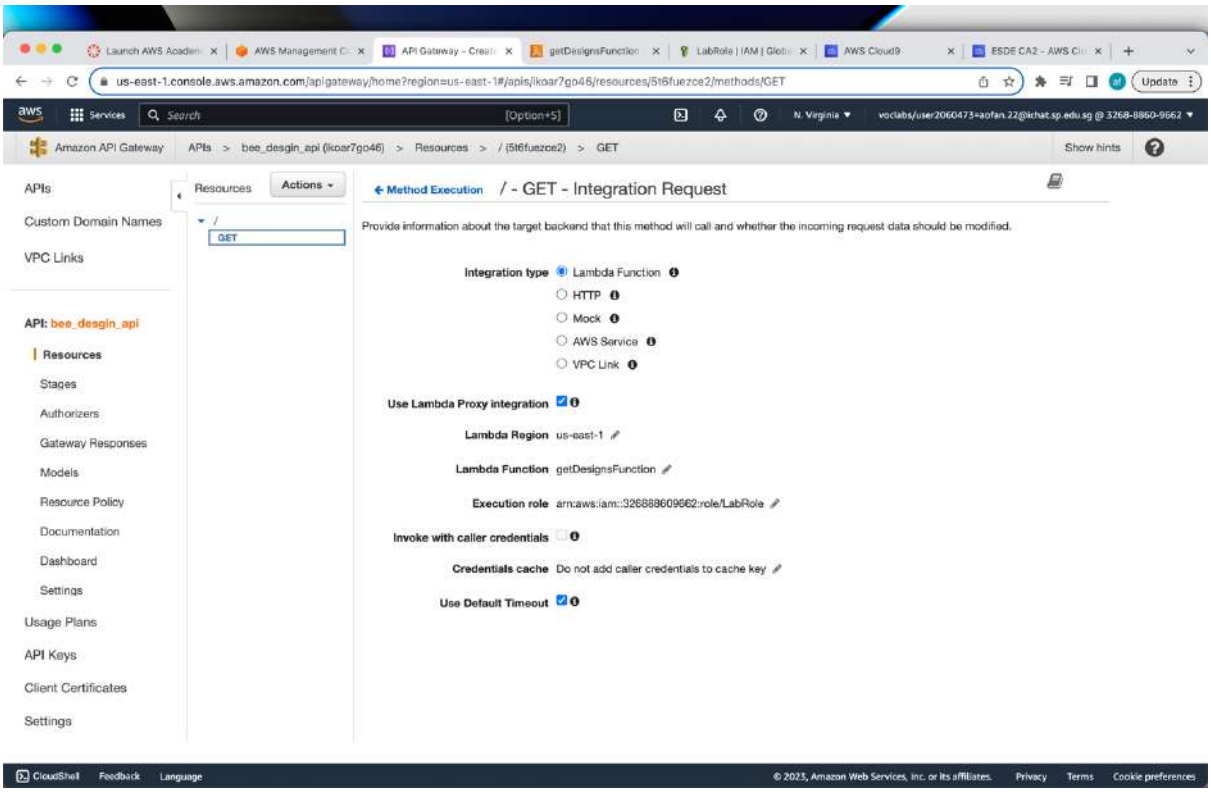
- Select Method Request.



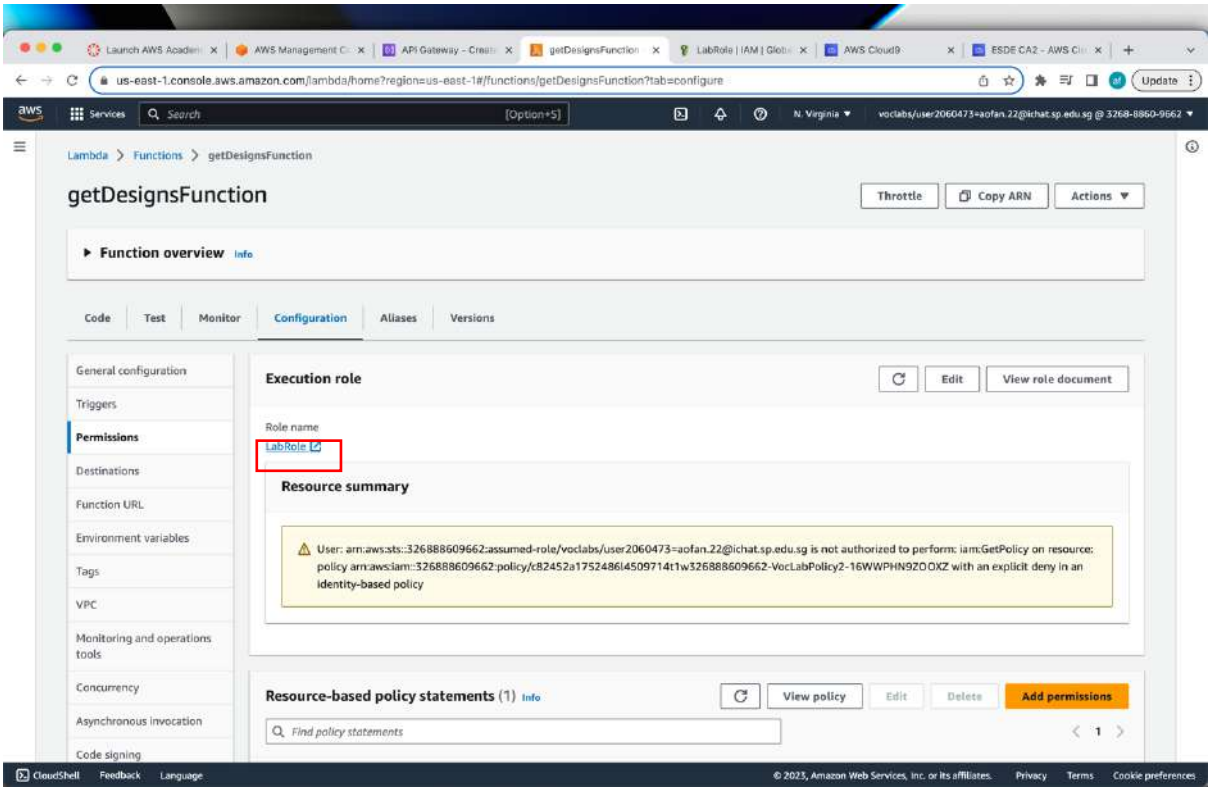
- Add a query string: fid and click on the tick to save. Get back to Method Execution via the link beside Actions button.



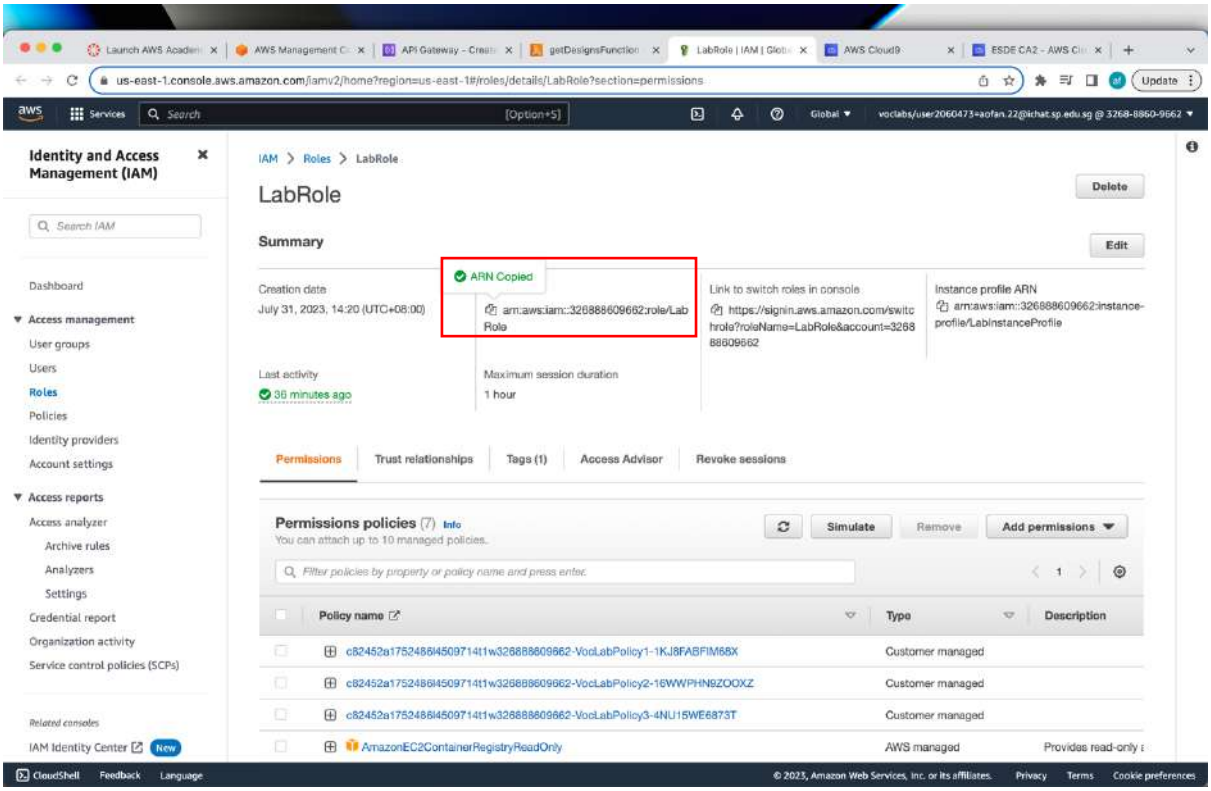
- Click on Integration Request. Set up as shown in the picture below. (Execution role can be obtained in the next two steps)



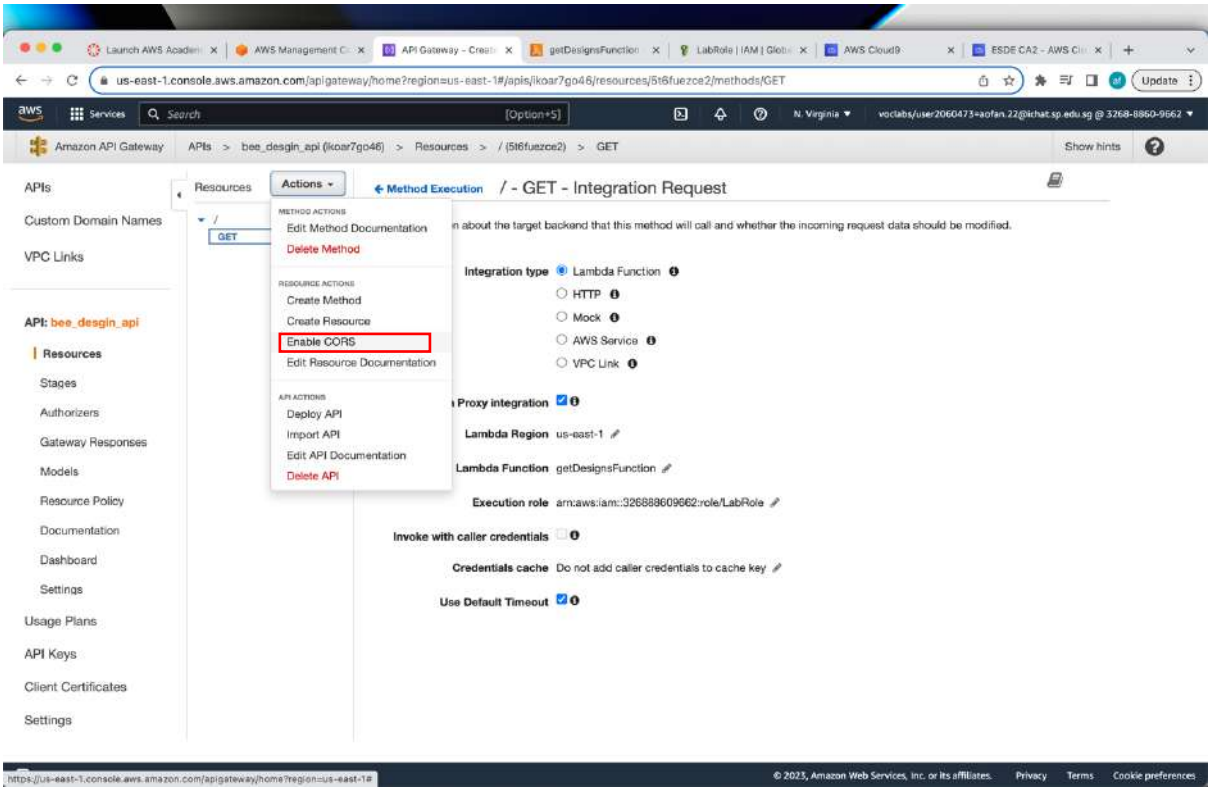
- Go to Lambda and click on LabRole.



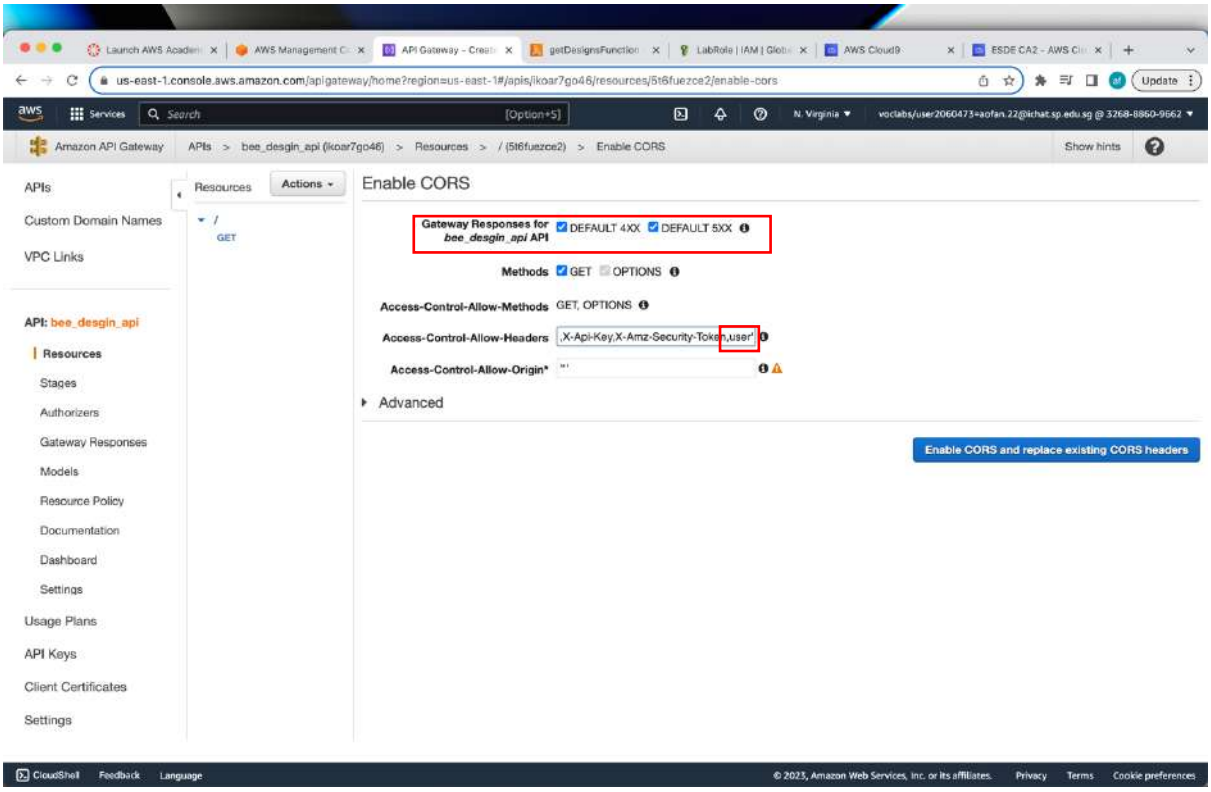
- Copy the ARN and paste it into Execution role.



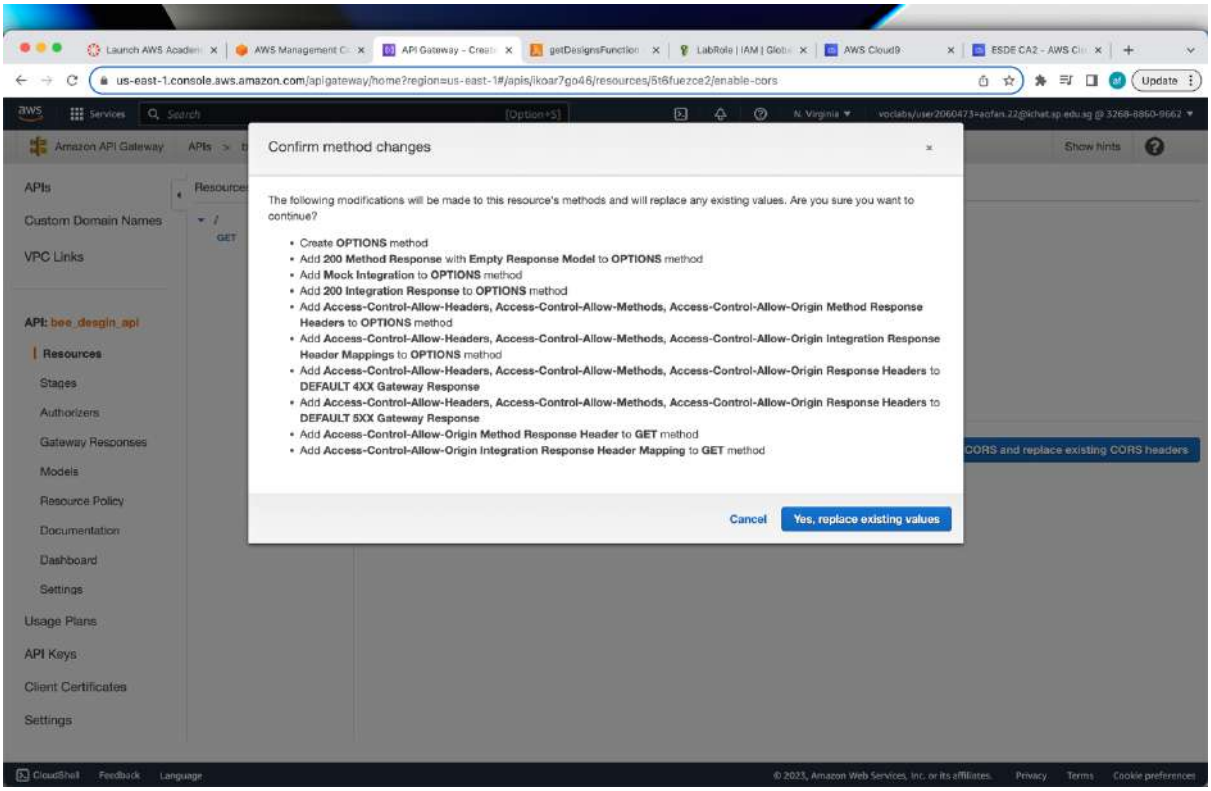
- After making changes in Integration Request, click on the Actions dropdown button and select Enable CORS.



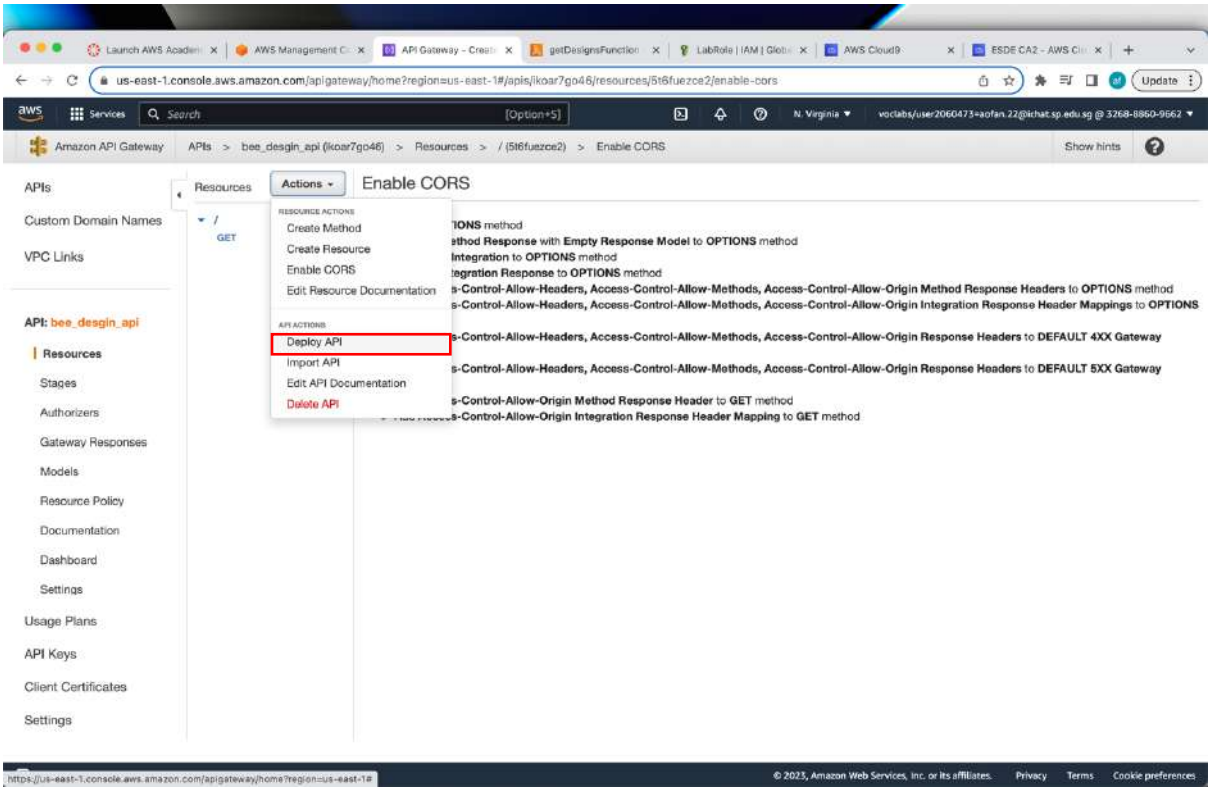
- Check the DEAFAULT 4XX and 5XX boxes and add ,user in Access-Control-Allow-Headers. Finally, click on Enable CORS and replace existing CROS headers.



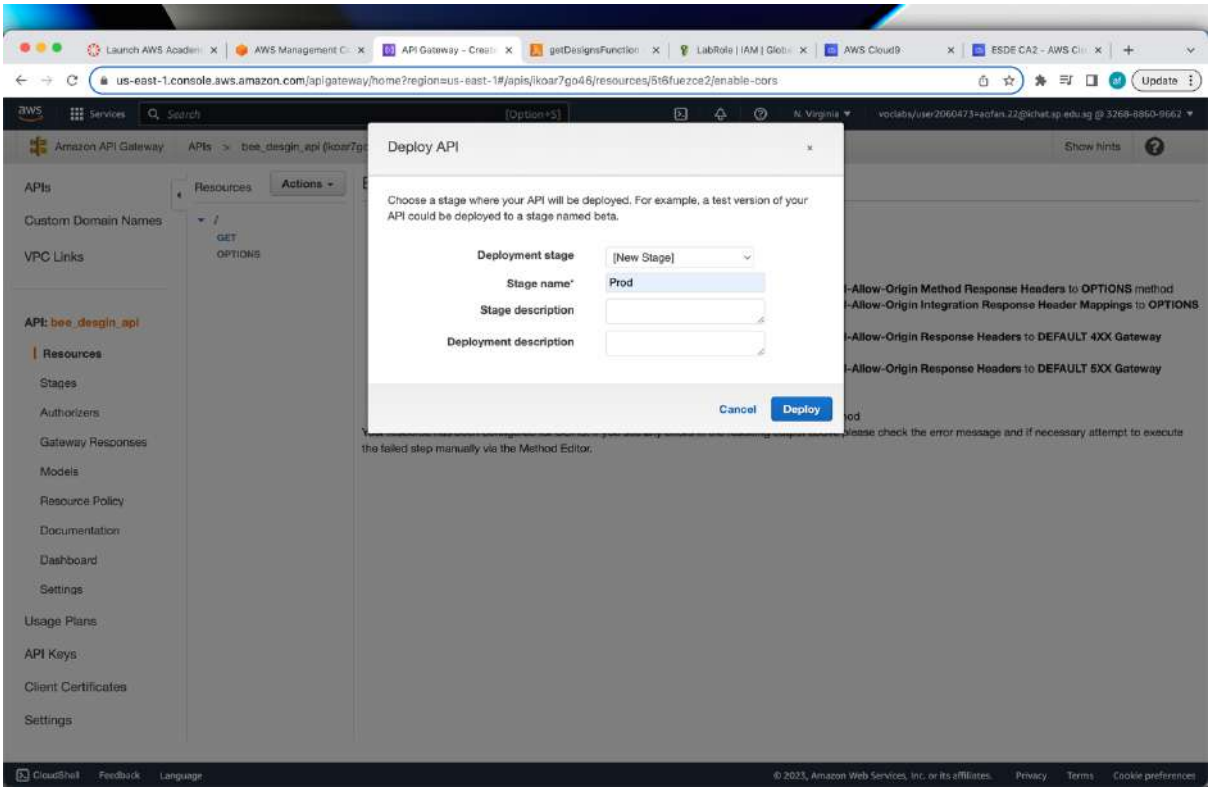
- Click Yes, replace existing values to confirm.



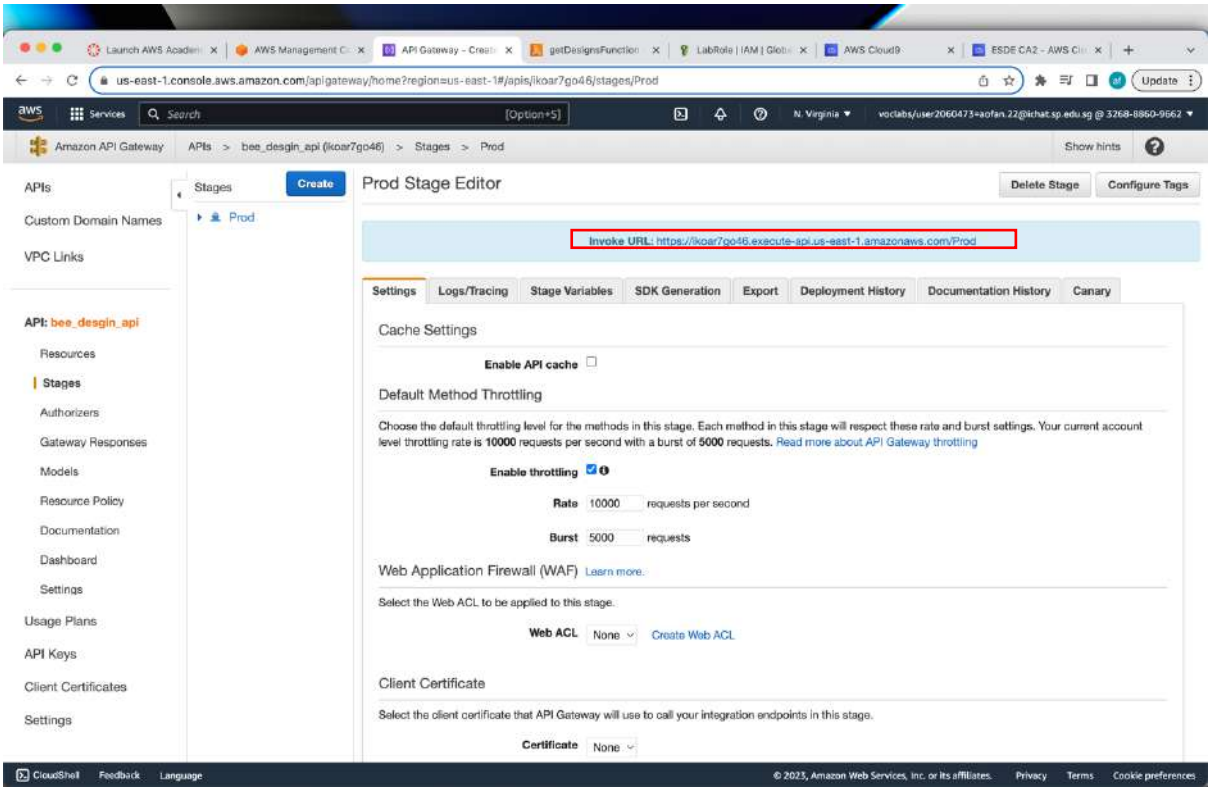
- Click on Actions dropdown button and select Deploy API.



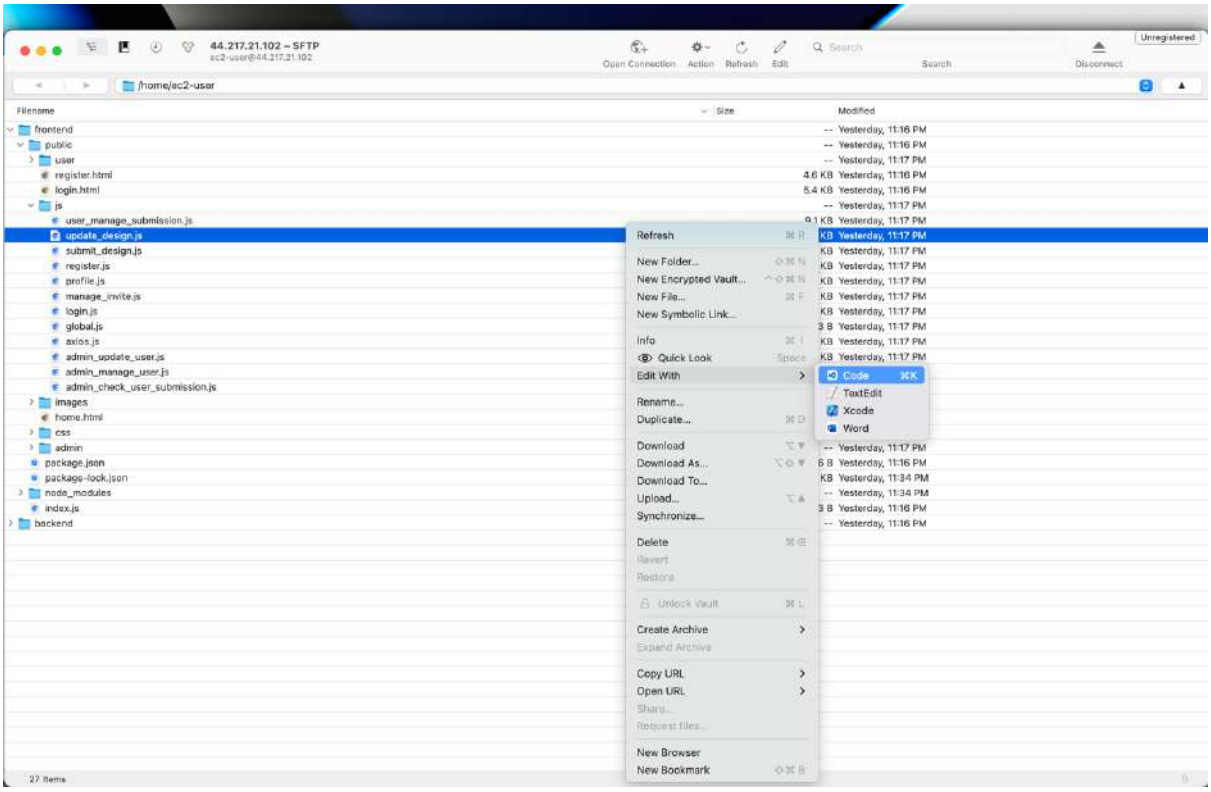
- Set up according to the illustration below and click Deploy.



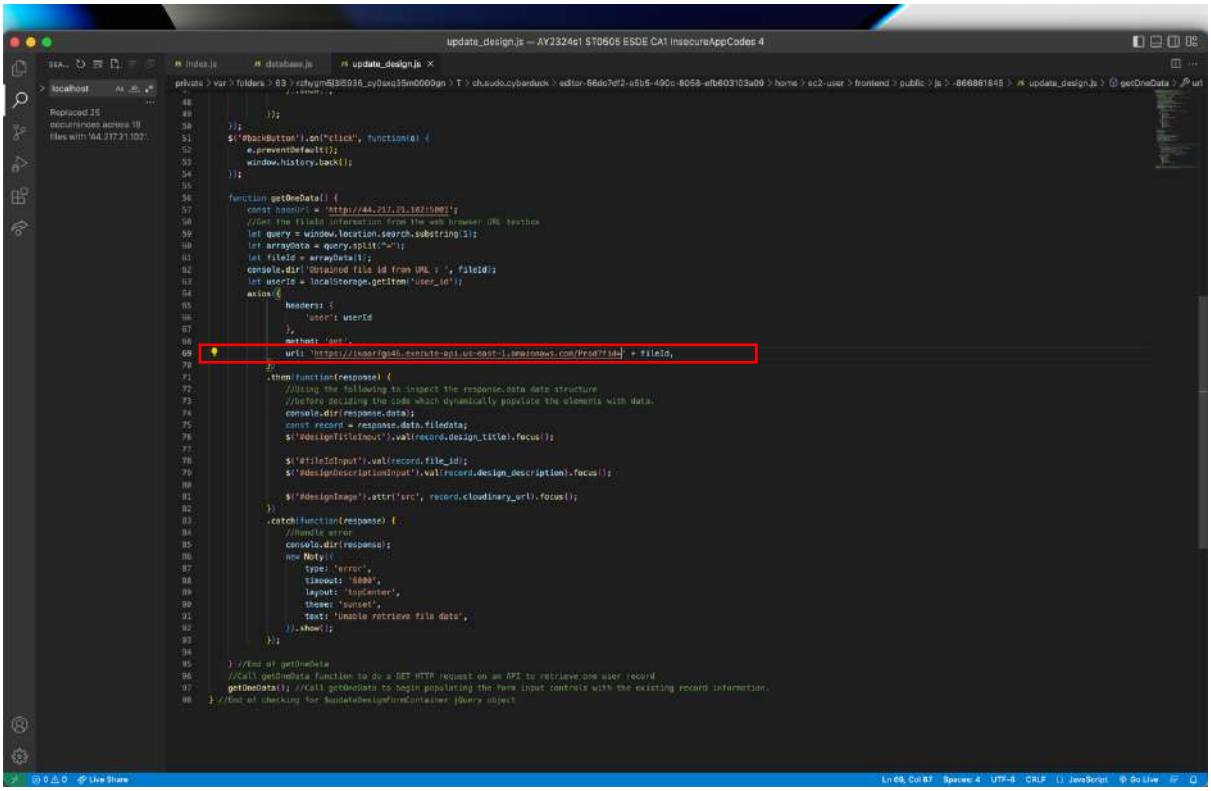
- Copy the Invoke URL and scroll to the bottom of the page and click Save Changes.



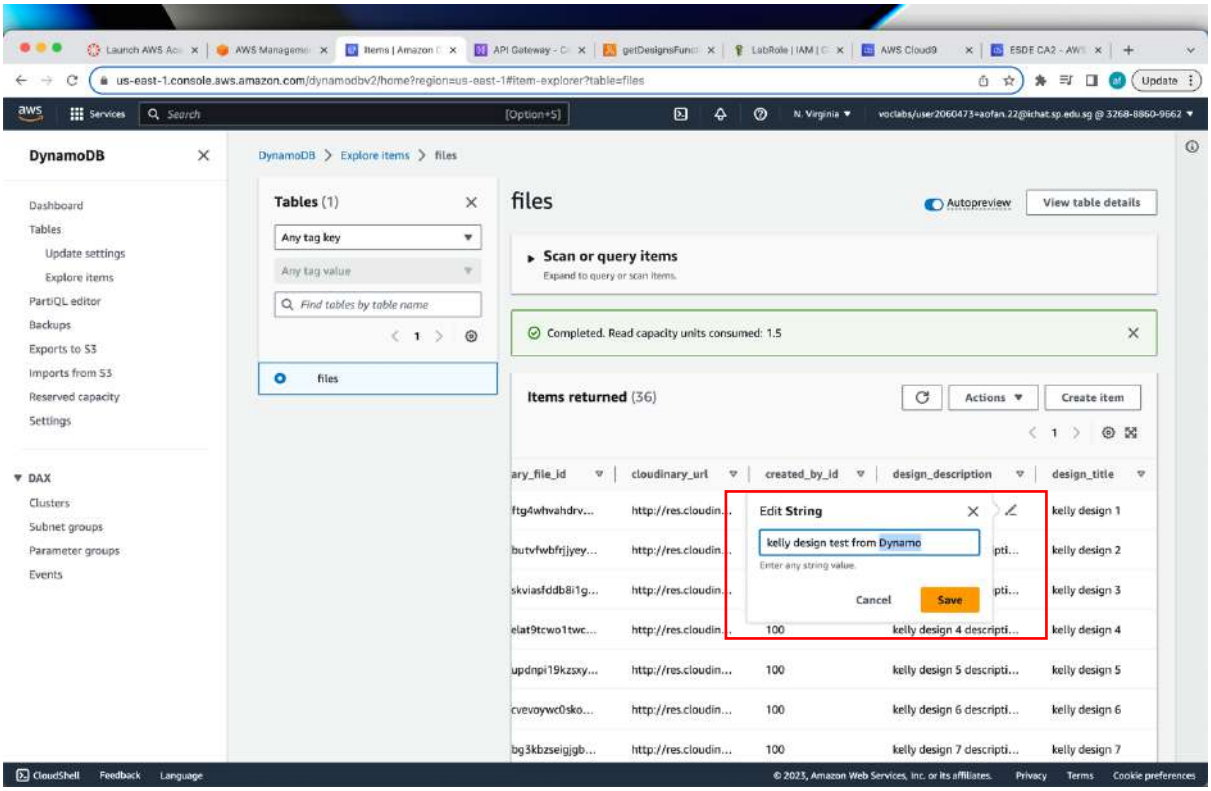
- Open Cyberduck and open update_design.js.



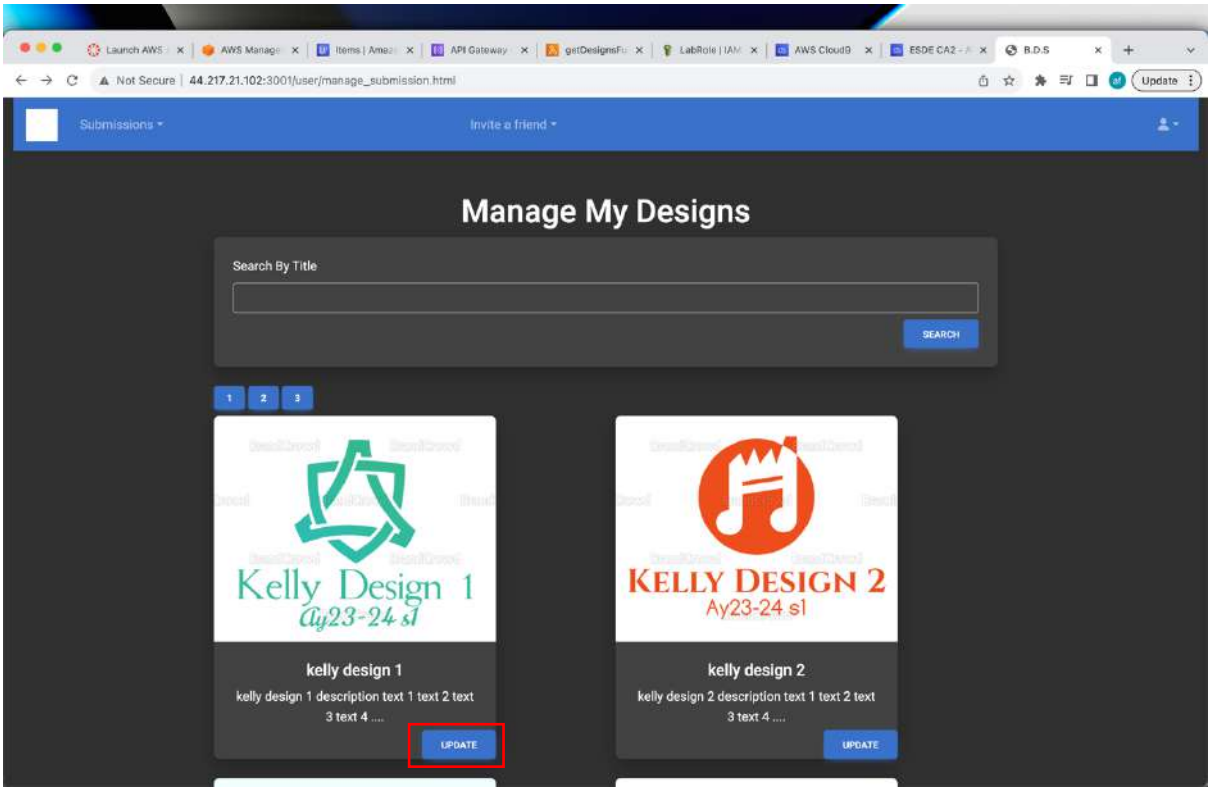
- Replace url with ‘Invoke URL?fid=’ + fileId and save the file. (line 69)



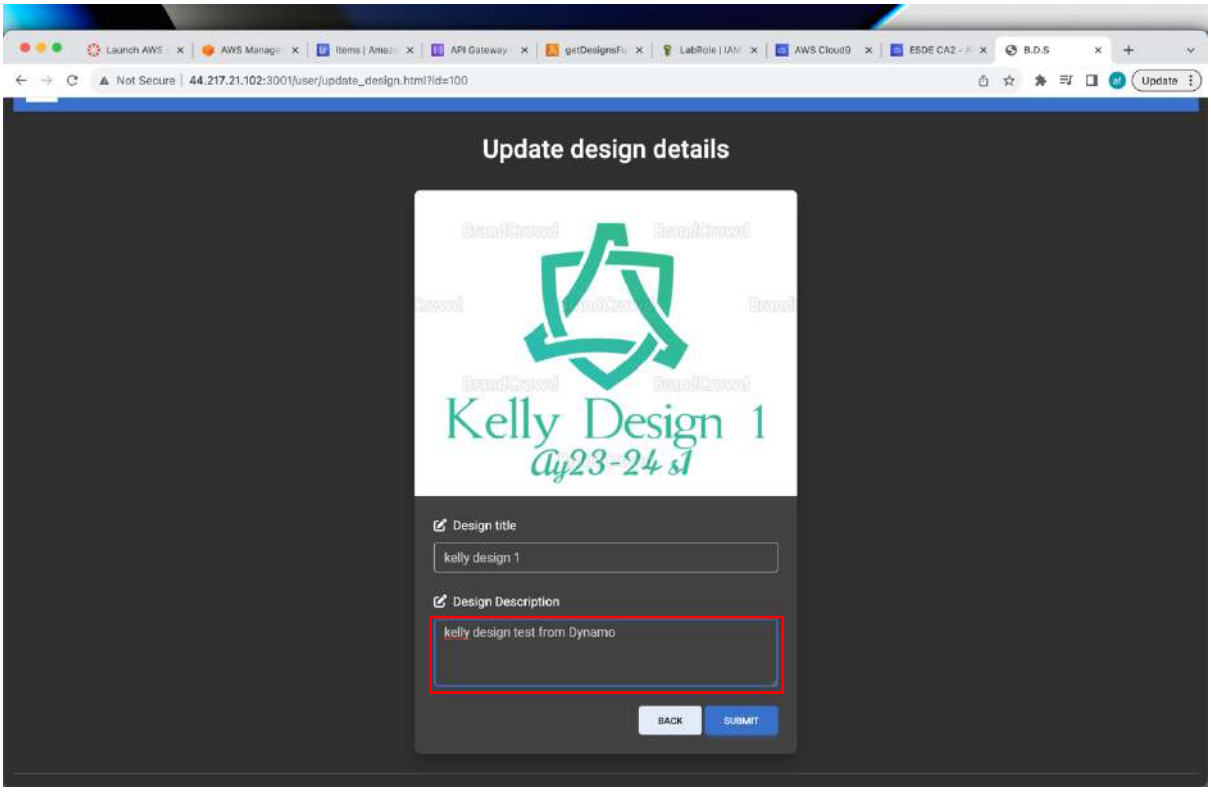
- Go to DynamoDB Table and edit the design_description of the file with the file_id of 100. Click Save.



- Launch the bee design app and login as kelly. Navigate to the update design page of the file with the file_id of 100. (which is kelly design 1)



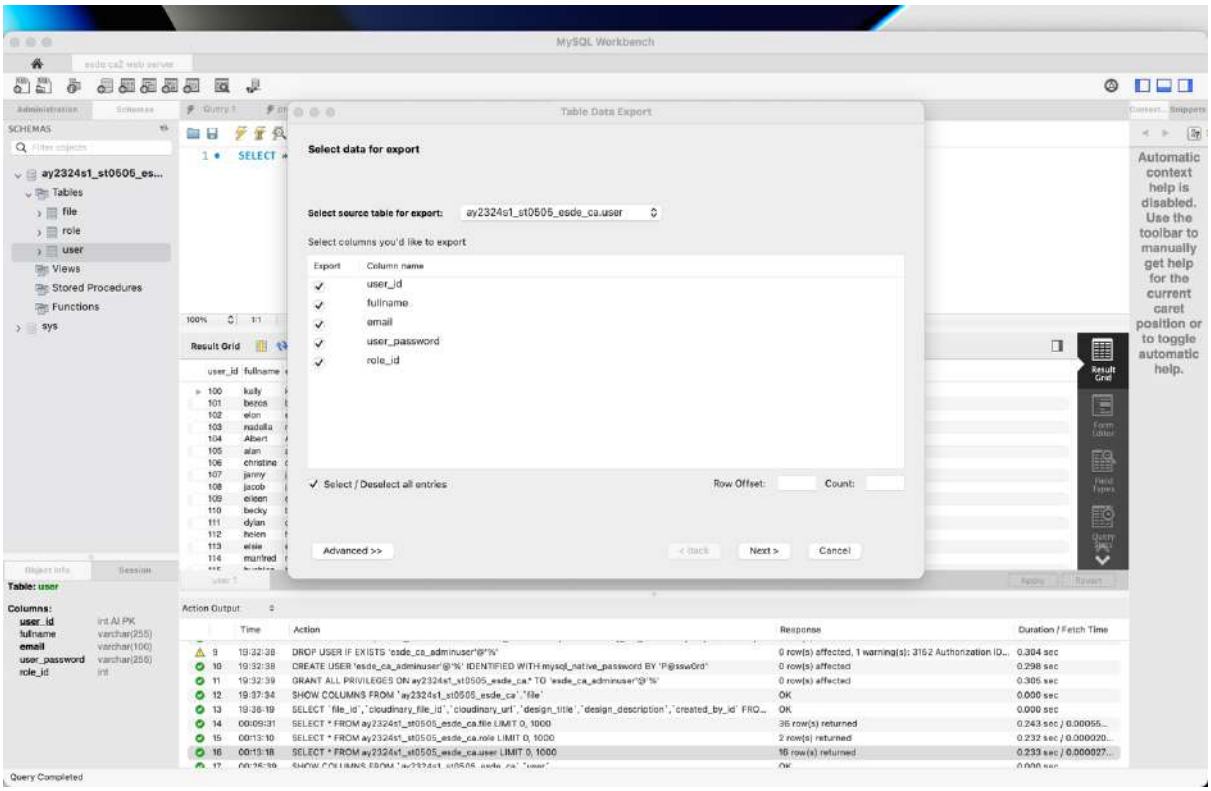
- If you see the edited design_description, it means that the API endpoint is functional.


Not Secure | 44.217.21.102:3001/user/update_design.html?id=100
Update
Update design details
Kelly Design 1
Design title
kelly design 1
Design Description
kelly design test from Dynamo
BACK
SUBMIT

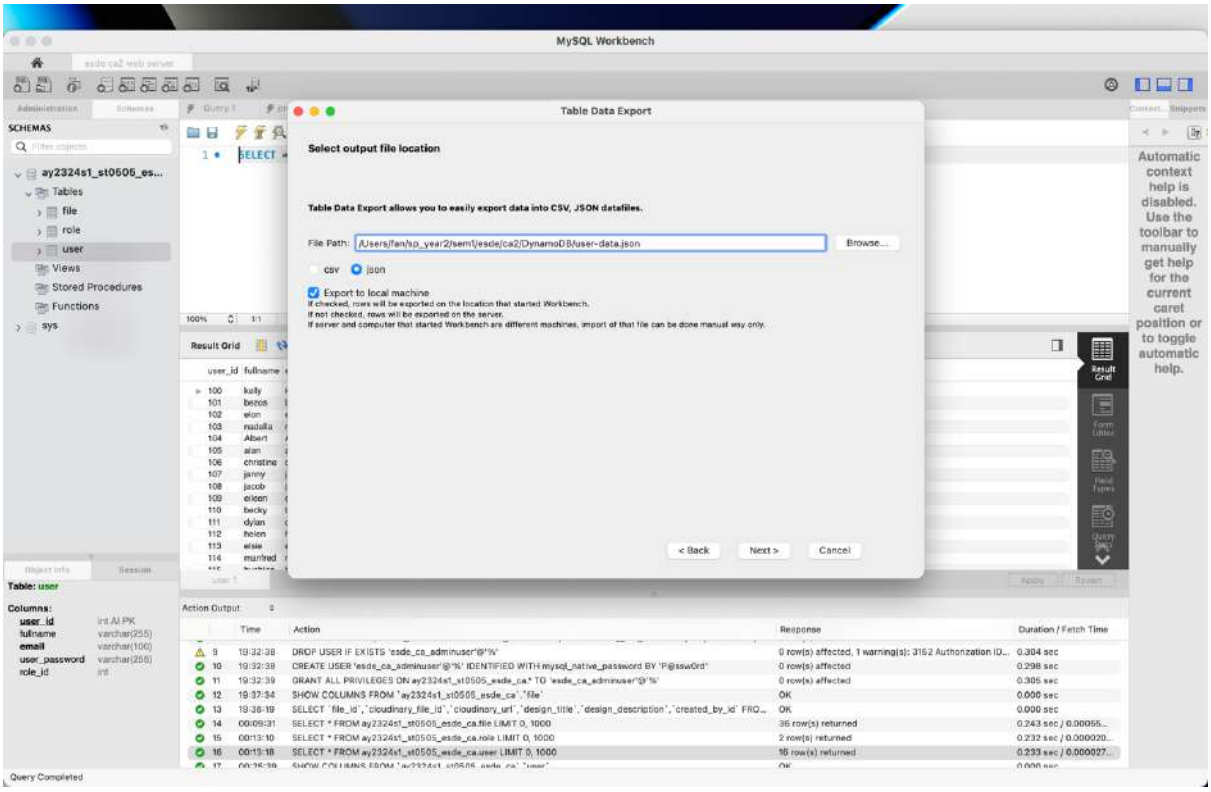
9. Additional Features

- Go to MySQLWorkbench, click on the esde ca2 web server connection. Right click on user located under ay2324s1_st0505_esde_ca, under Tables and select Data Table Export Wizard.

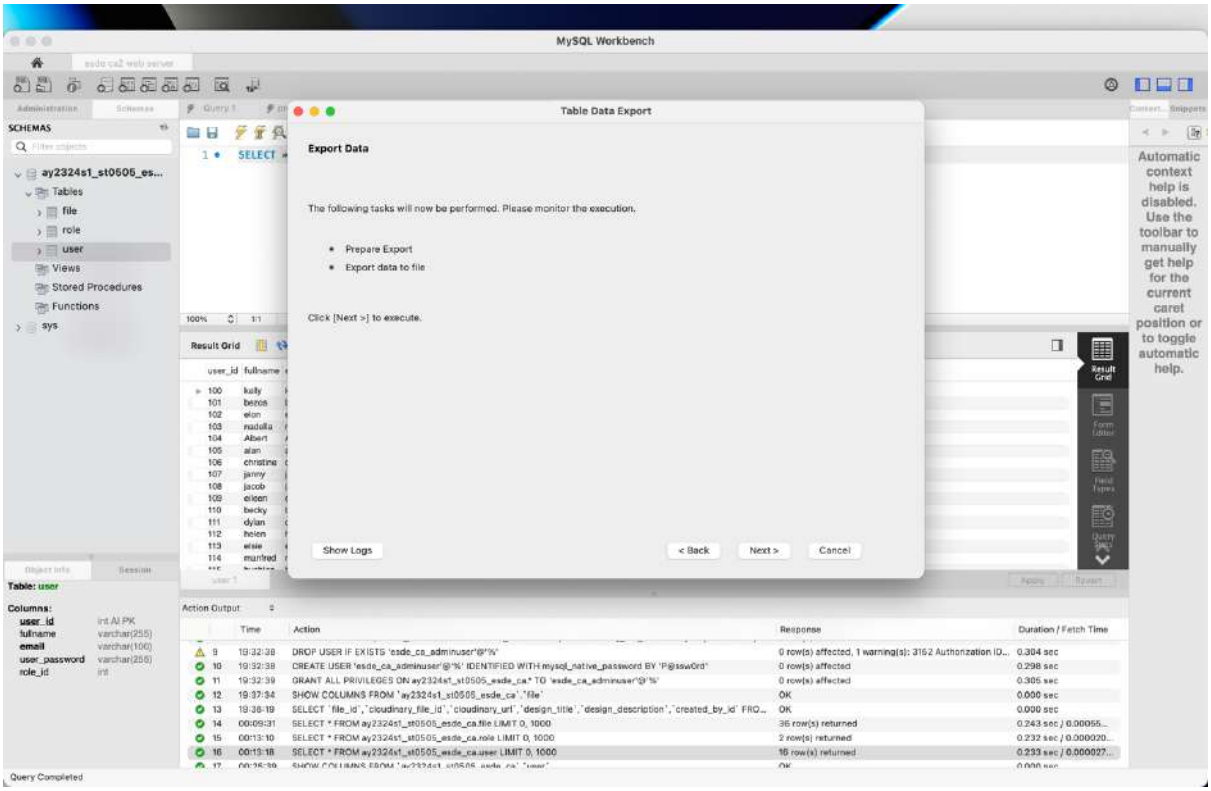
- On Table Data Export tab, click Next >.

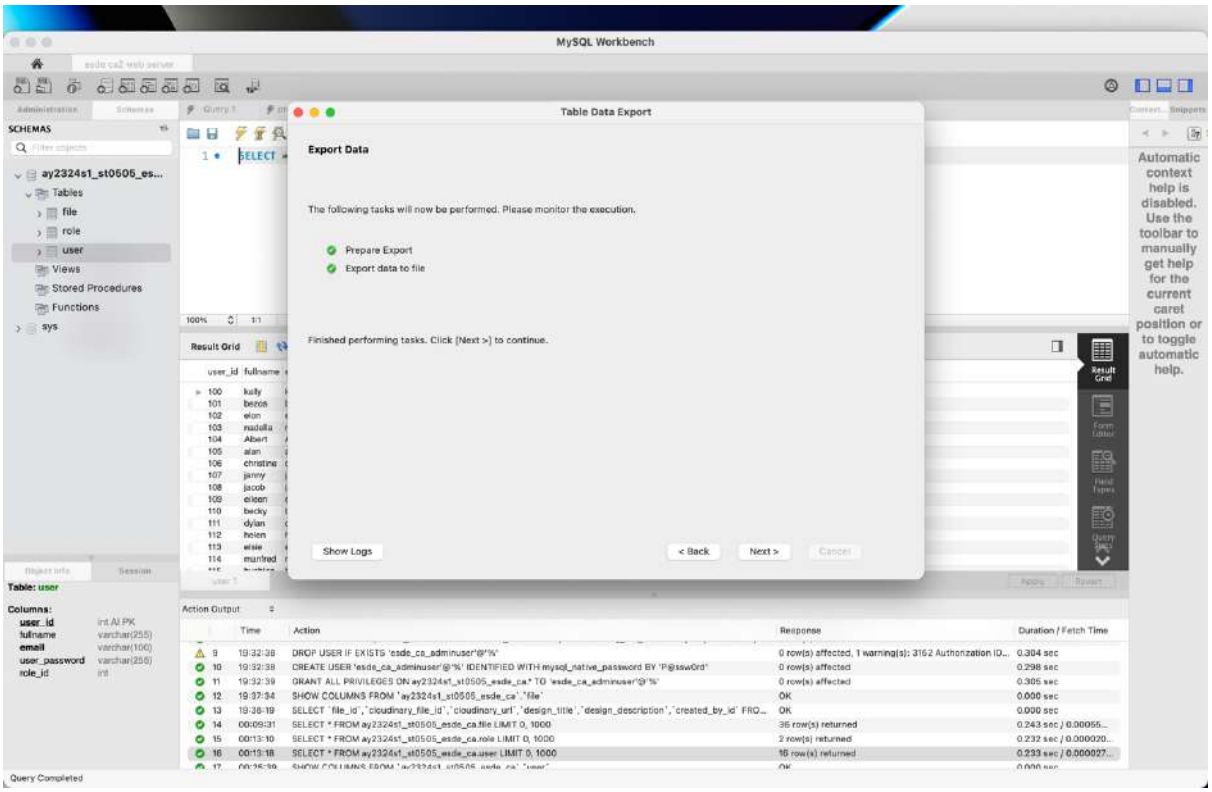


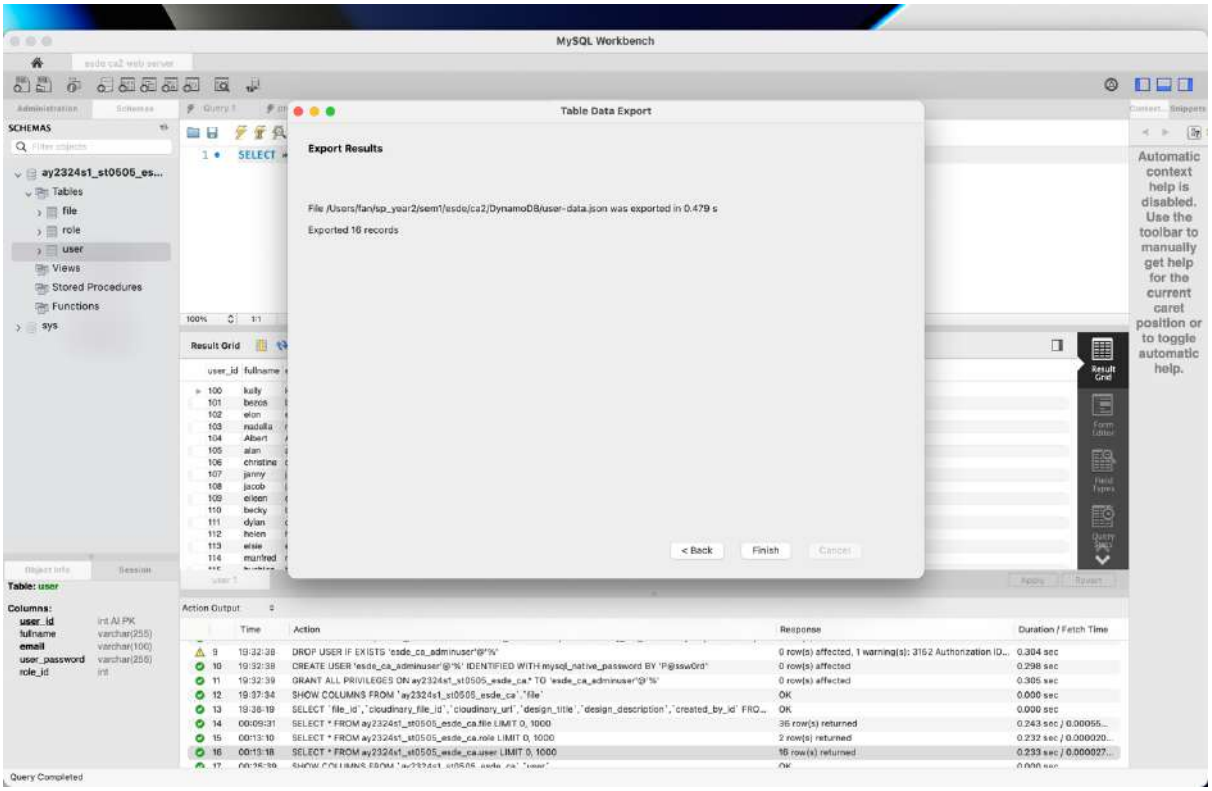
- Select json radio button and click on Browse... Name the file user-data.json and click Save.



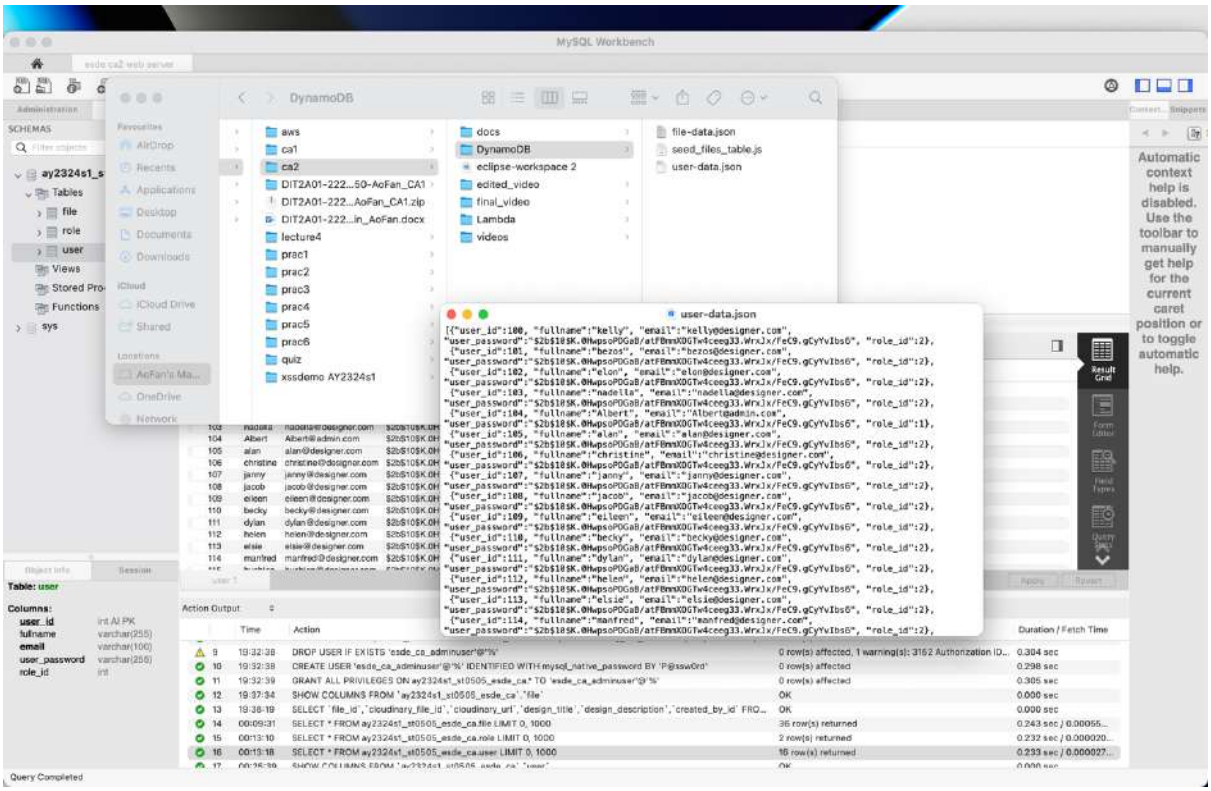
- Click Next > followed by Next > followed by Next >. Lastly Click Finish.




MySQL Workbench
Table Data Export
Export Data
The following tasks will now be performed. Please monitor the execution.
Prepare Export
Export data to file
Finished performing tasks. Click [Next >] to continue.
Show Logs
< Back
Next >
Cancel
SCHEMAS
ay2324s1_st0505_es...
Tables
file
role
user
Views
Stored Procedures
Functions
sys
Result Grid
Action Output
Time
Action
Response
Duration / Fetch Time
Table: user
Columns:
user_id
fullname
email
user_password
role_id
Query Completed


MySQL Workbench
Table Data Export
Export Results
File /Users/fan/sp_year2/sem1/esde/ca2/DynamoDB/user-data.json was exported in 0.479 s
Exported 16 records
< Back
Finish
Cancel
SCHEMAS
ay2324s1_st0505_es...
Tables
file
role
user
Views
Stored Procedures
Functions
sys
Result Grid
Action Output
Time
Action
Response
Duration / Fetch Time
Table: user
Columns:
user_id
fullname
email
user_password
role_id
Query Completed

- The file exported should look like this in TextEdit.



- Navigate to DynamoDB via AWS Management Console. Click Create table. Set Table name to be users, partition key name to be user_id, select Number from the dropdown list and scroll to the bottom of the page and click on Create table. Head to DynamoDB Tables tab to check the newly created table.

- Navigate to Cloud9

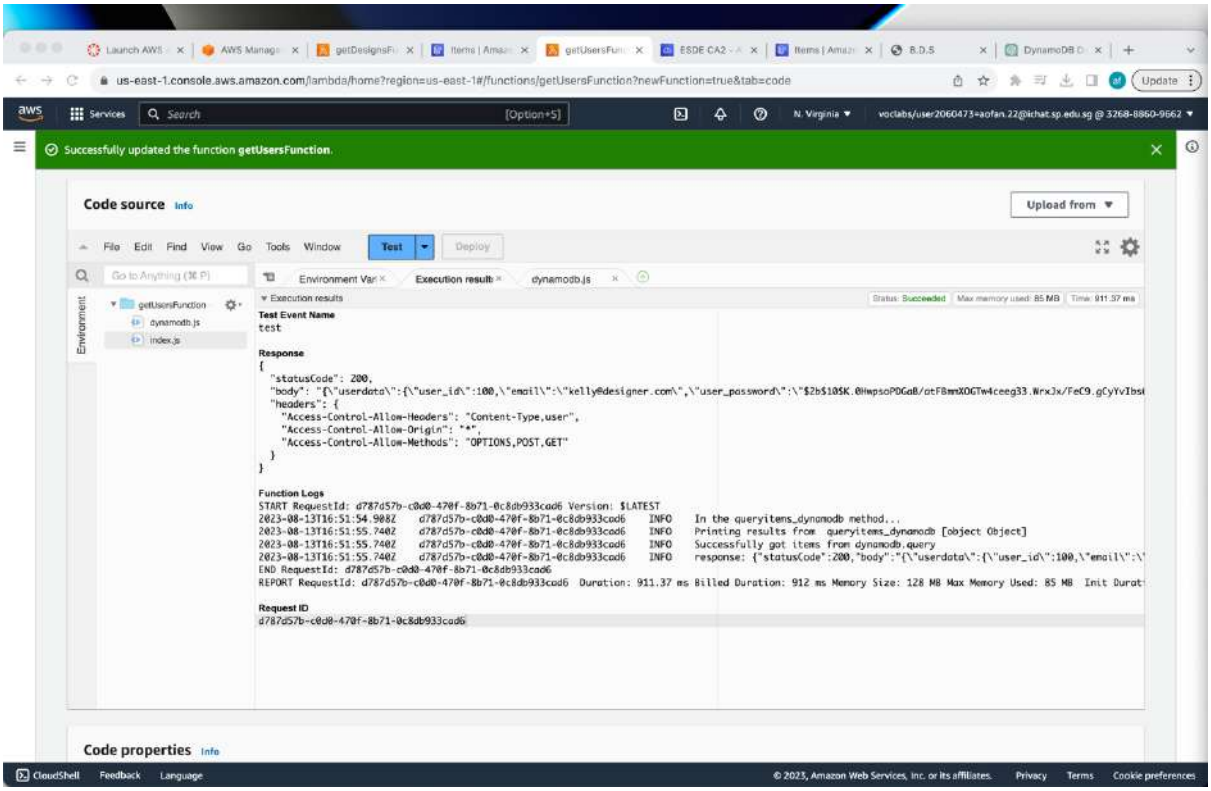
- Enter the IDE and run seed_users_tables.js to populate the table

- Go to Lambda and create a function called getUsersFunction, similar to the setup of getDesignsFunction.

- After creating the function, go to Cloud9 and open IDE and change the index.js file such that it will work for getUsersFunction .

- Run zip function.zip index.js dynamodb.js. Followed by aws lambda update-function-code --function-name getUsersFunction --zip-file fileb://function.zip.

- Go to Lambda and test the function.



- Next, set up an endpoint and use it in bee design app. (Refer to 8. Setting up API endpoint for Lambda function using API Gateway)