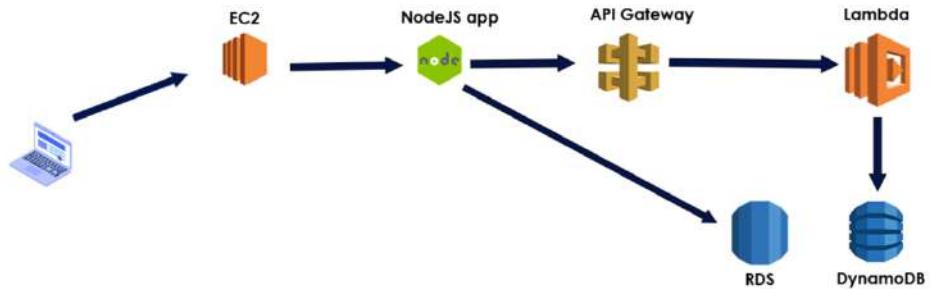


1. Architecture Diagram



2. Creating an EC2 instance

- Login into AWS academy account.
- Navigate to AWS Academy Learner Lab [45760] via the dashboard.

The screenshot shows the AWS Academy dashboard. On the left, there is a sidebar with icons for Account, Dashboard (which is highlighted with a red box), Courses, Calendar, Inbox, History, and Help. The main area is titled "Dashboard" and contains two cards. The first card, "AWS Academy Cloud Developing [... ACDv1EN-11994]", is purple. The second card, "AWS Academy Learner Lab [45760] ALLv1-45760", is dark gray and is also highlighted with a red box. To the right of the cards, there is a "Recent feedback" section listing three knowledge checks with 100 out of 100 points. At the bottom right, there is a "View Grades" button.

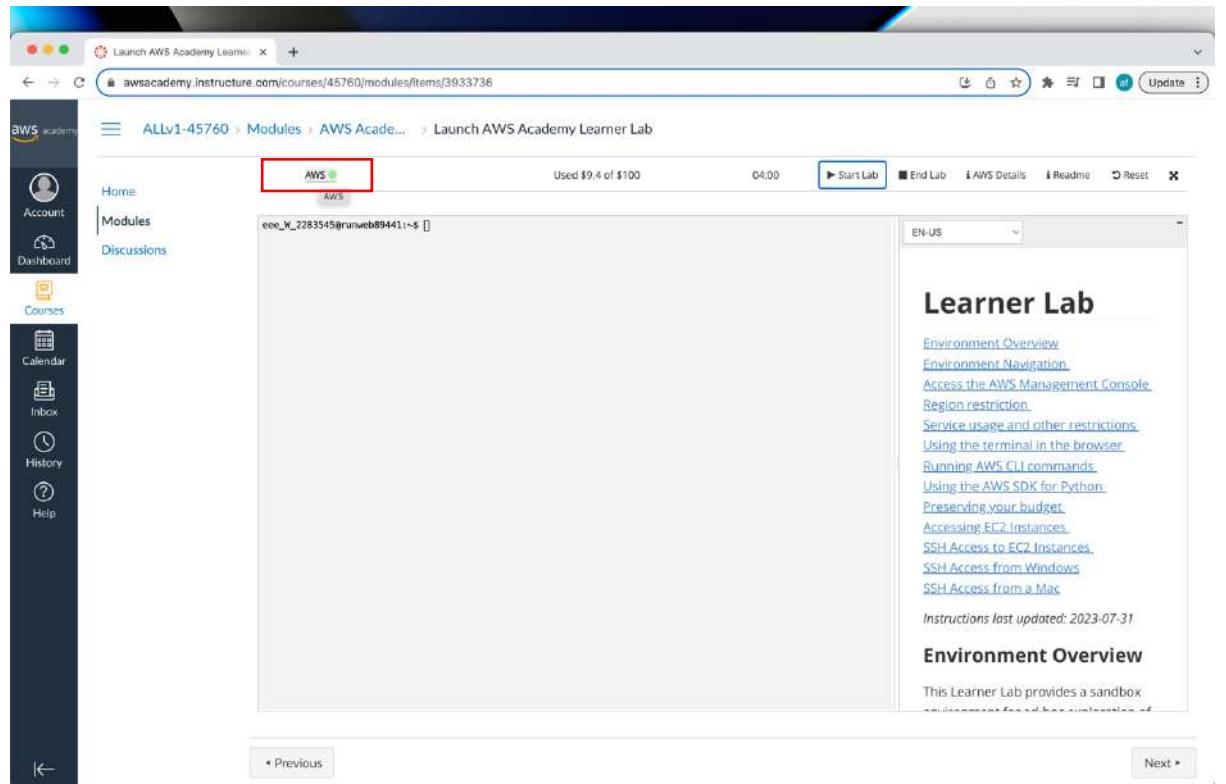
- Click on Modules section and launch the AWS Academy Learner Lab.

The screenshot shows the AWS Academy course modules page. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The 'Modules' icon is highlighted with a red box. The main content area shows a list of modules under 'AWS Academy Learner Lab'. One item, 'Launch AWS Academy Learner Lab', is also highlighted with a red box.

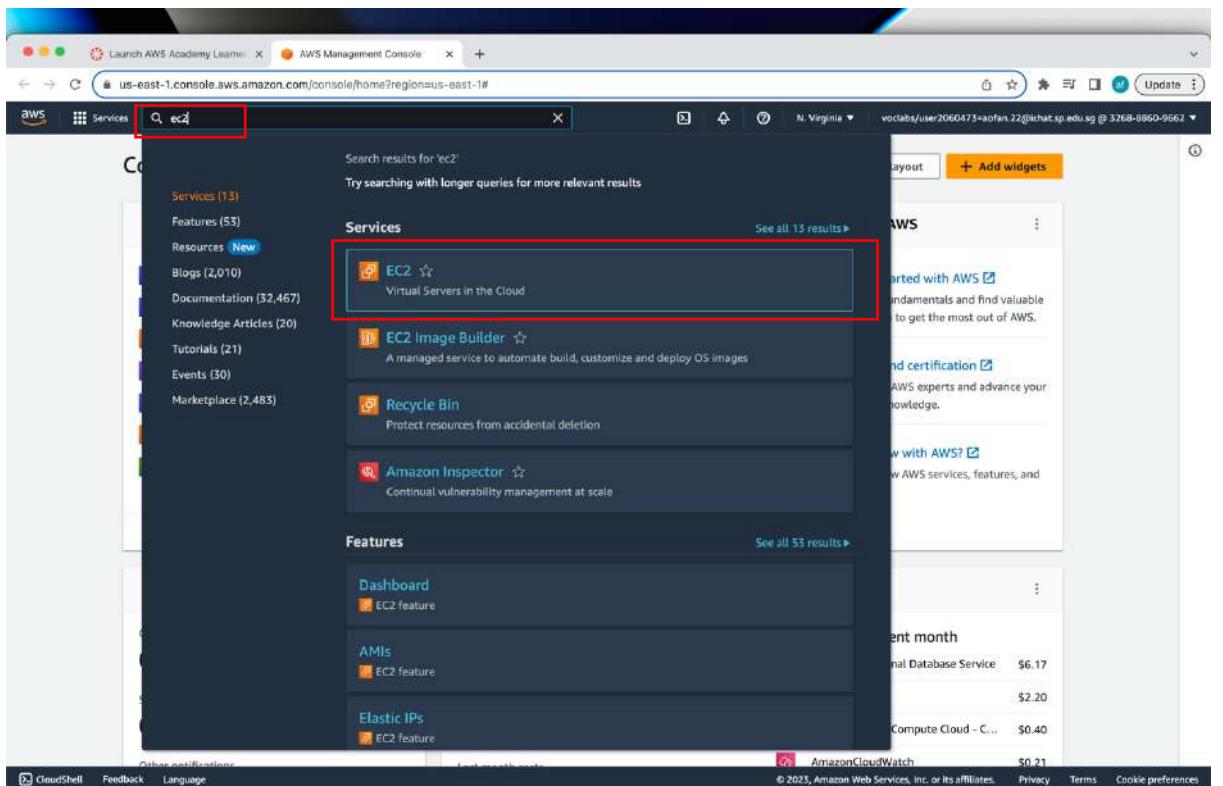
- Start the Lab.

The screenshot shows the AWS Academy Learner Lab interface. At the top, there's a toolbar with 'Start Lab' (highlighted with a red box), 'End Lab', 'AWS Details', 'Readme', and 'Reset'. Below the toolbar is a terminal window showing the command 'echo_H_22835458runweb89441:-\$'. To the right of the terminal is a sidebar titled 'Learner Lab' with sections like 'Environment Overview', 'Environment Navigation', and 'Access the AWS Management Console'. At the bottom, there's a message 'Waiting for heapanalytics.com...' and navigation buttons for 'Previous' and 'Next'.

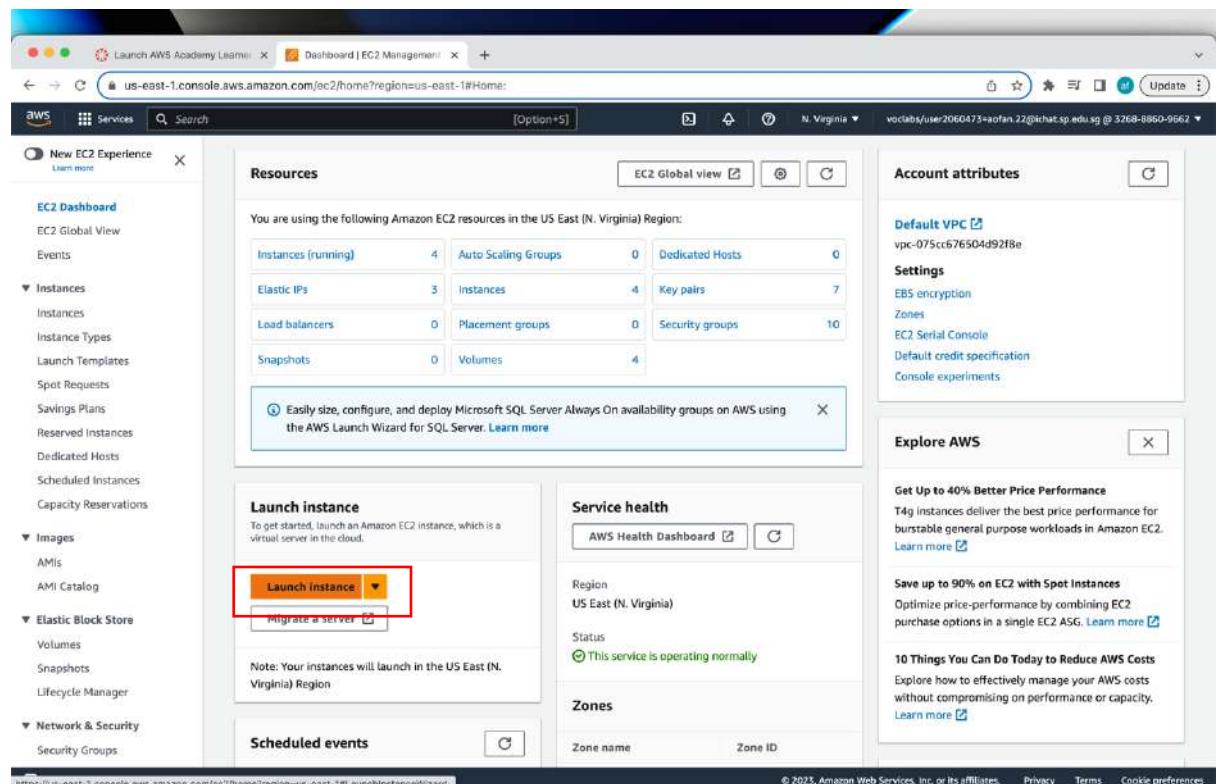
- Click on AWS to navigate to the AWS Console.



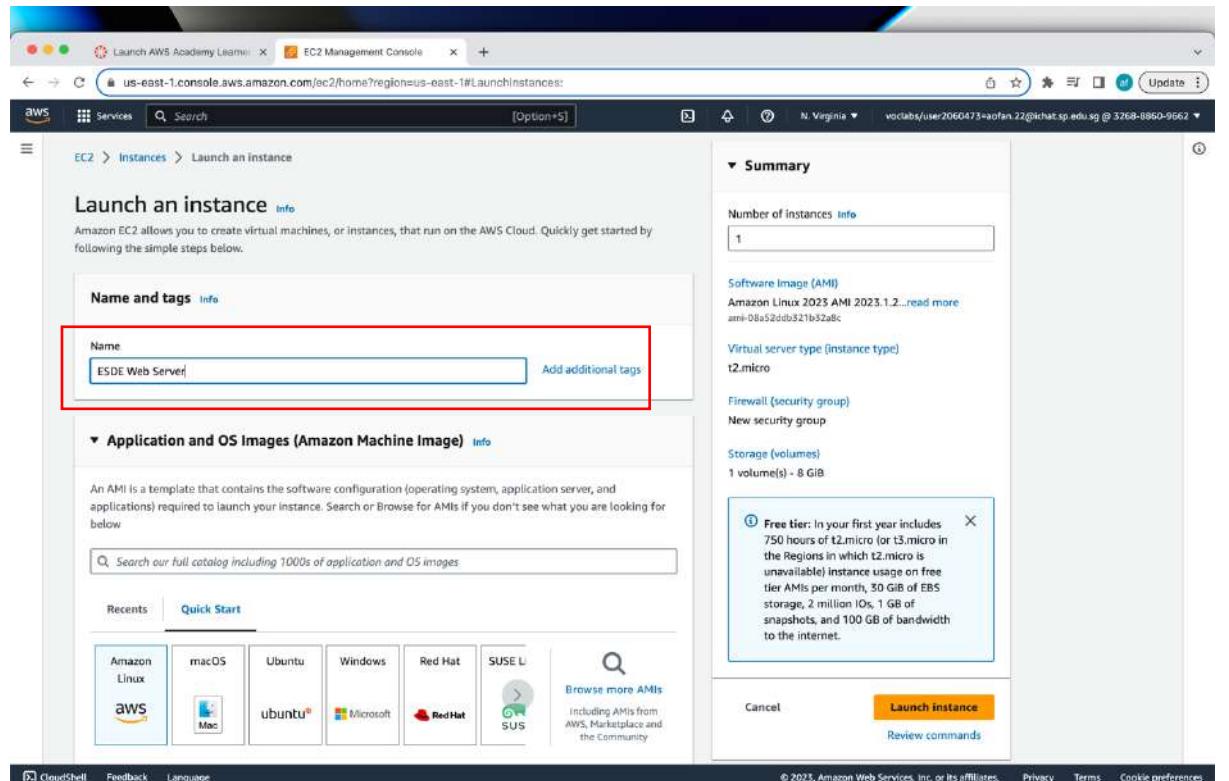
- Search “EC2” in AWS Console to go to EC2 Dashboard.



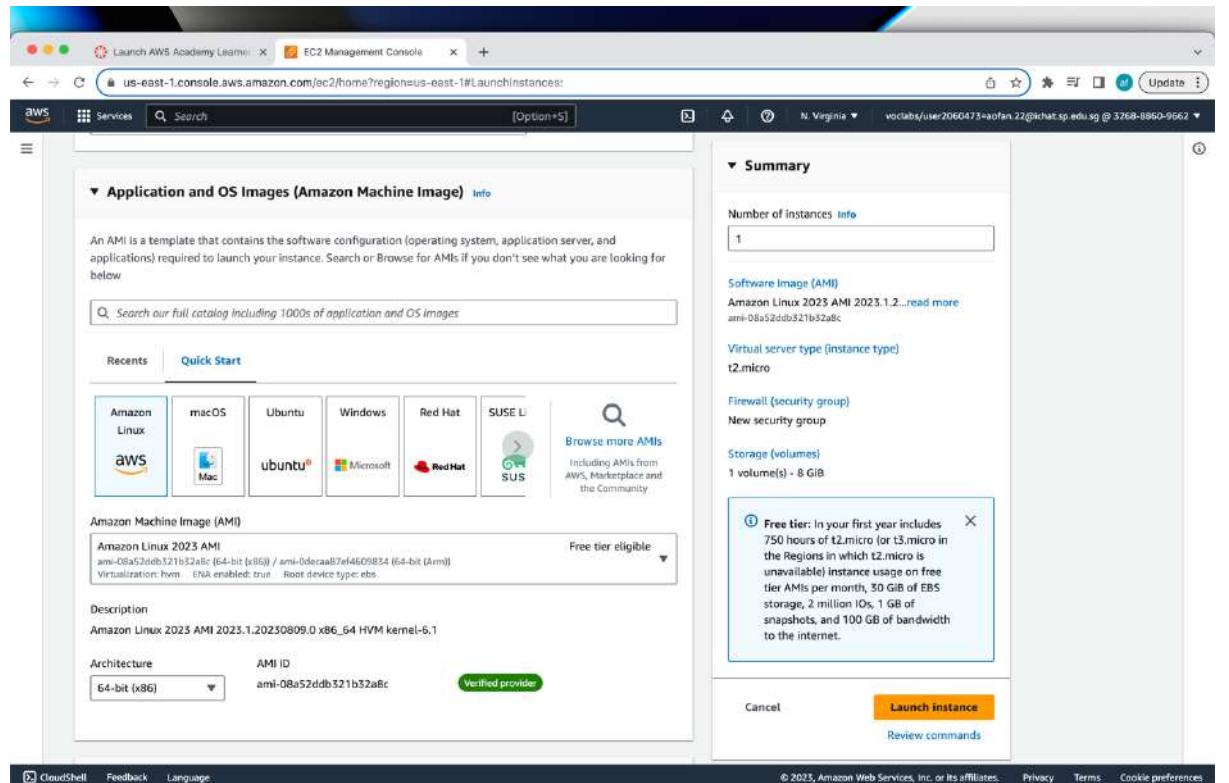
- Click on Launch instance to create a new EC2 instance.



- For Name and tags: ESDE Web Server.



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



- For Instance type: Free tier eligible.

Instance type

t2.micro

Free tier eligible

Key pair (login)

Select

Network settings

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel Launch instance Review commands

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

Create key pair

Key pair name: esde_ca2_key_pair

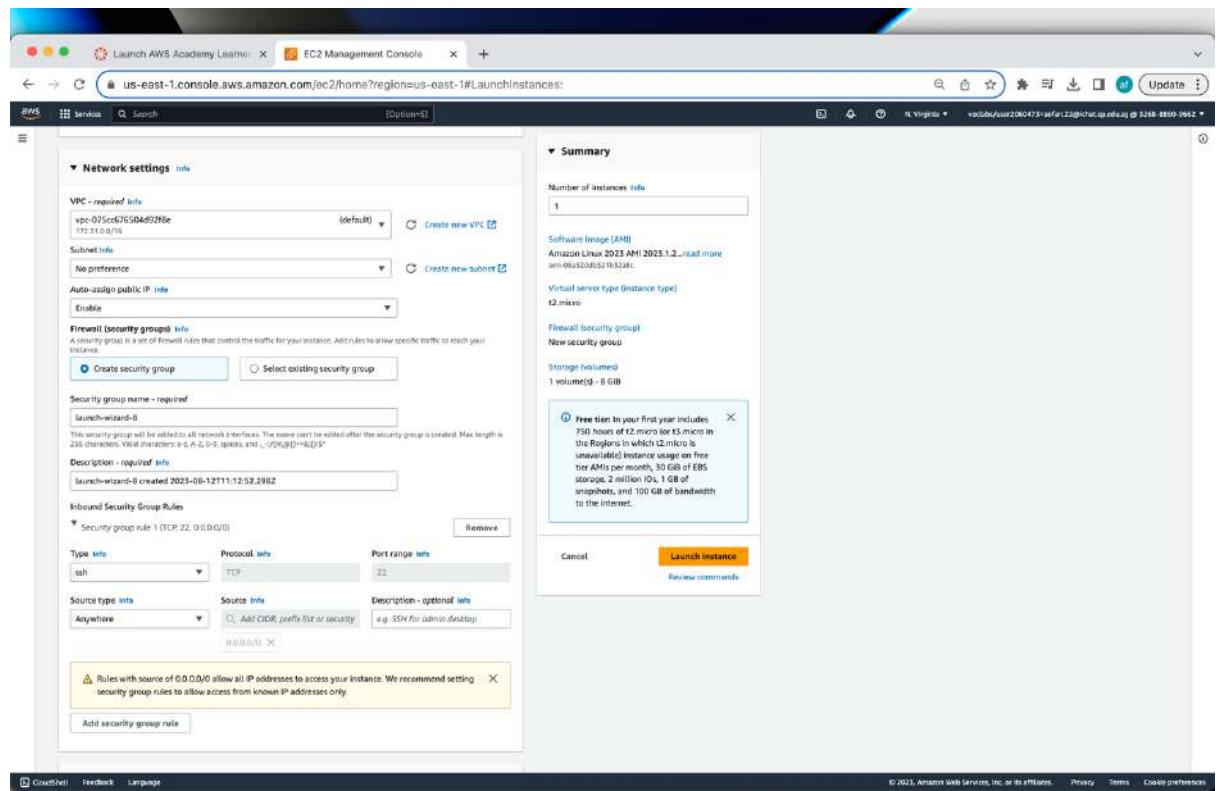
Key pair type: RSA

Private key file format: .pem

Warning: When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance.

Cancel Create key pair

- For Network settings:



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

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
test6	i-0795e1e5612a567fa	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-237-192
esde-Webserv...	i-05916cd43b39ad052	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-234-176
ESDE Web Ser...	i-0f44d275b981ea95a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-205-13

Instance: i-0f44d275b981ea95a (ESDE Web Server)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0f44d275b981ea95a (ESDE Web Server)	Public IPv4 address 54.205.133.66 [open address]	Private IPv4 addresses 172.31.80.128
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-205-13-66.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-80-128.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-80-128.ec2.internal	

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

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
test6	i-0795e1e5612a567fa	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-237-192
esde-Webserv...	i-05916cd43b39ad052	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-34-234-176
ESDE Web Ser...	i-0f44d275b981ea95a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-205-13

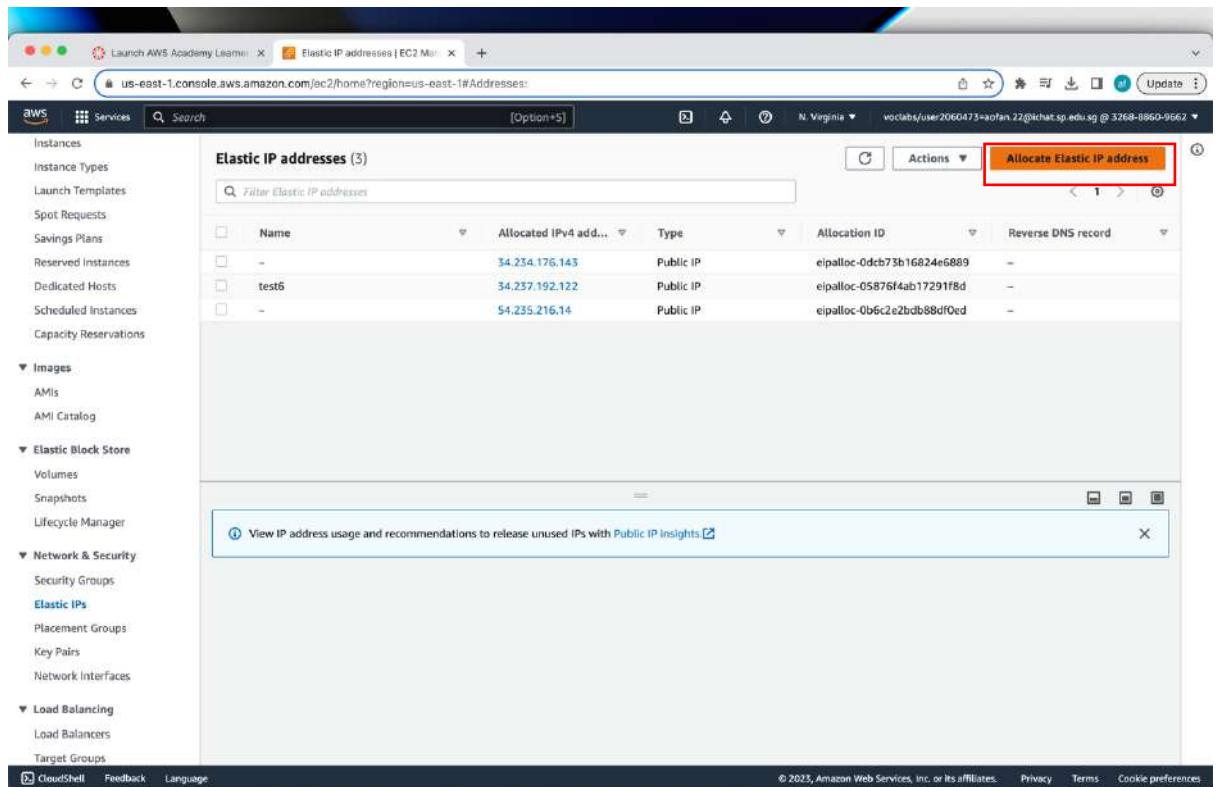
Instance: i-0f44d275b981ea95a (ESDE Web Server)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID i-0f44d275b981ea95a (ESDE Web Server)	Public IPv4 address 54.205.133.66 [open address]	Private IPv4 addresses 172.31.80.128
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-205-13-66.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-80-128.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-80-128.ec2.internal	

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



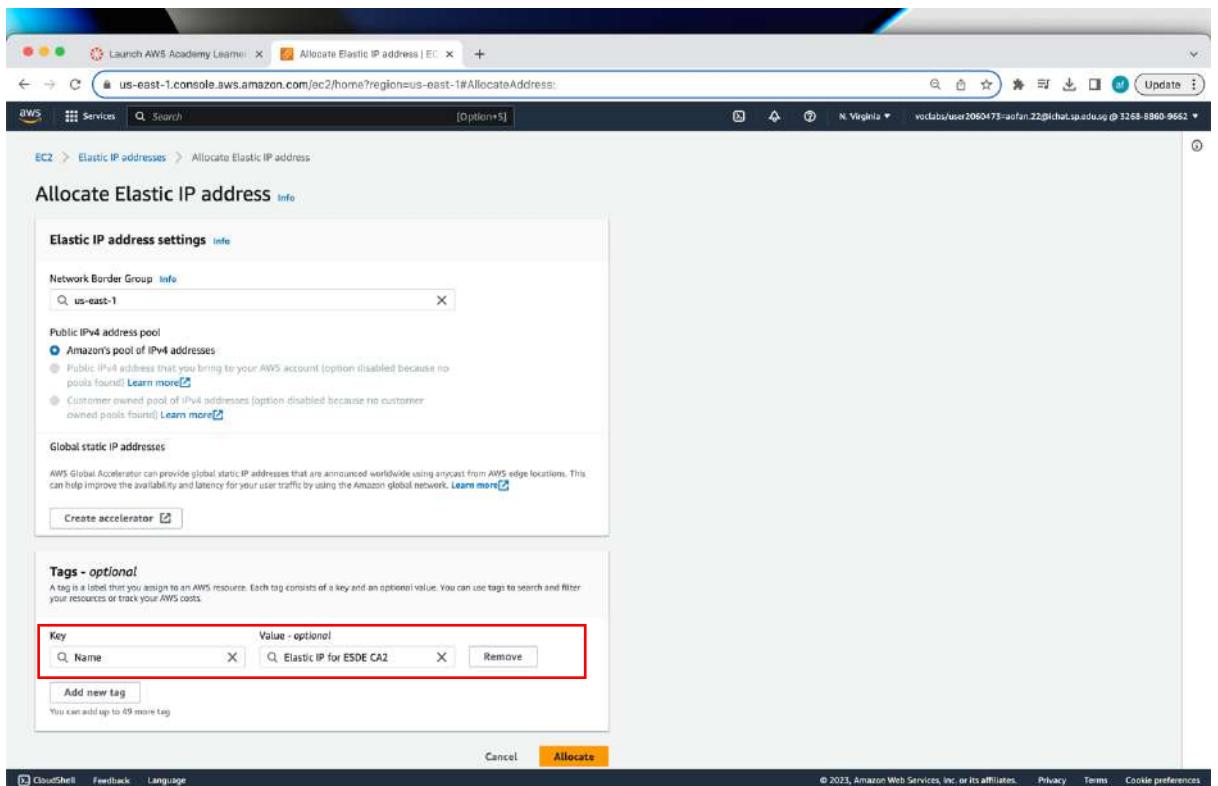
The screenshot shows the AWS Management Console interface for managing Elastic IP addresses. The left sidebar navigation includes 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'Images', 'AMIs', 'AMI Catalog', 'Elastic Block Store', 'Volumes', 'Snapshots', 'Lifecycle Manager', 'Network & Security', 'Security Groups', 'Elastic IPs' (which is selected), 'Placement Groups', 'Key Pairs', 'Network Interfaces', 'Load Balancing', 'Load Balancers', and 'Target Groups'. The main content area is titled 'Elastic IP addresses (3)' and lists three entries:

Name	Allocated IPv4 add...	Type	Allocation ID	Reverse DNS record
-	34.234.176.143	Public IP	eipalloc-0dcb73b16824e6889	-
test6	34.237.192.122	Public IP	eipalloc-05876f4ab17291f8d	-
-	54.235.216.14	Public IP	eipalloc-0b6c2e2bdb88df0ed	-

At the top right of the main content area, there is a 'Actions' dropdown menu with an 'Allocate Elastic IP address' option, which is highlighted with a red box. Below the table, a callout box with a blue arrow points to the 'View IP address usage and recommendations to release unused IPs with Public IP Insights' link.

- 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.

Elastic IP addresses (1/4)

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID
test6	54.234.176.143	Public IP	eipalloc-0ccb73b16824e6889	-	i-05916cd43b59ad032
<input checked="" type="checkbox"/> Elastic IP for ESDE CA2	44.217.21.102	Public IP	eipalloc-05876f4ab17291f8d	-	-
-	54.235.216.14	Public IP	eipalloc-0b6c2e2bdb88df0ed	-	-

View IP address usage and recommendations to release unused IPs with Public IP insights.

44.217.21.102

Summary Tags

Summary

Allocated IPv4 address	Type	Allocation ID	Reverse DNS record
44.217.21.102	Public IP	eipalloc-0ab5ce06f0a770e47	-
-	Association ID	Scope	Associated instance ID
-	-	-	Private IP address

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

Elastic IP address allocated successfully.
Elastic IP address 44.217.21.102 / Elastic IP for ESDE CA2.

Elastic IP addresses (1/4)

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID
test6	54.234.176.143	Public IP	eipalloc-0ccb73b16824e6889	-	i-05916cd43b59ad032
<input checked="" type="checkbox"/> Elastic IP for ESDE CA2	44.217.21.102	Public IP	eipalloc-05876f4ab17291f8d	-	-
-	54.235.216.14	Public IP	eipalloc-0b6c2e2bdb88df0ed	-	-

View IP address usage and recommendations to release unused IPs with Public IP insights.

44.217.21.102

Summary Tags

Summary

Allocated IPv4 address	Type	Allocation ID	Reverse DNS record
44.217.21.102	Public IP	eipalloc-0ab5ce06f0a770e47	-
-	Association ID	Scope	Associated instance ID
-	-	-	Private IP address

Actions Associate this Elastic IP address

Associate Elastic IP address

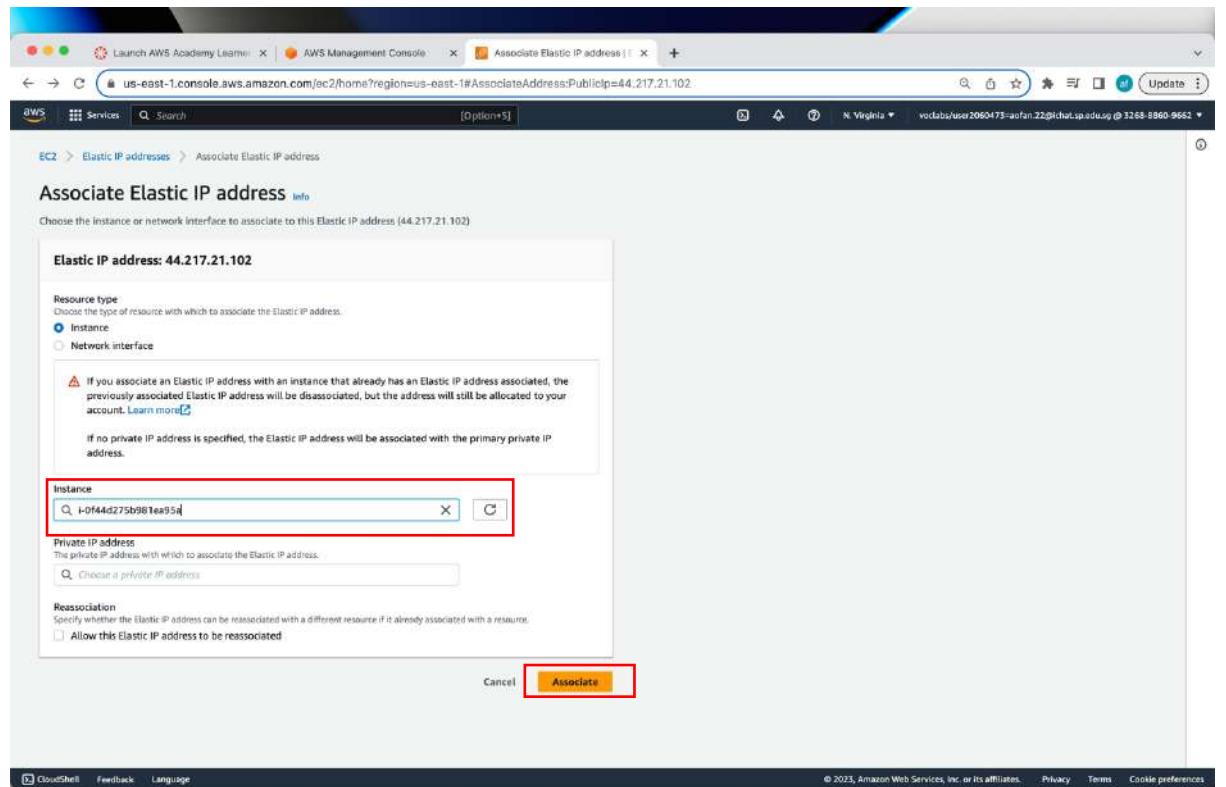
Release Elastic IP addresses

Update reverse DNS

Enable transfers

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- 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.

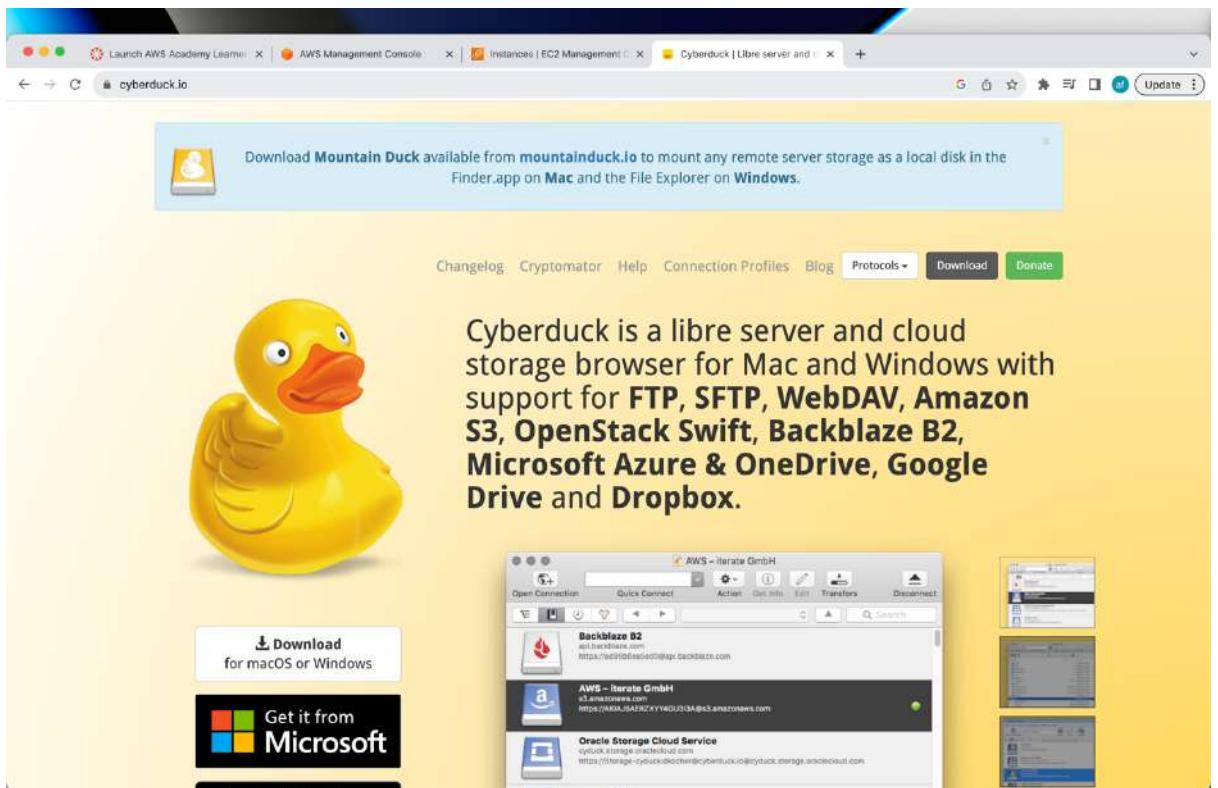
The screenshot shows the AWS Management Console EC2 Instances page. On the left, there's a navigation sidebar with various services like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch-Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
esde-Webserv...	i-05916cd43b19ad832	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-34-254-176-184.co...	54.234.176...
ESDE Web ...	i-0f44d275b981ea95a	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-44-217-21-102.co...	44.217.21...
aws-cloud9-ES...	i-0c32712f60bd19f91	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1c	ec2-18-212-216-197.co...	18.212.216...
test6	i-0795e1e5612a567fa	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-34-237-192-122.co...	34.237.192...
aws-cloud9-ES...	i-02e0ef9eba23f4f3b	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-5-88-150-184.com...	5.88.150.184...

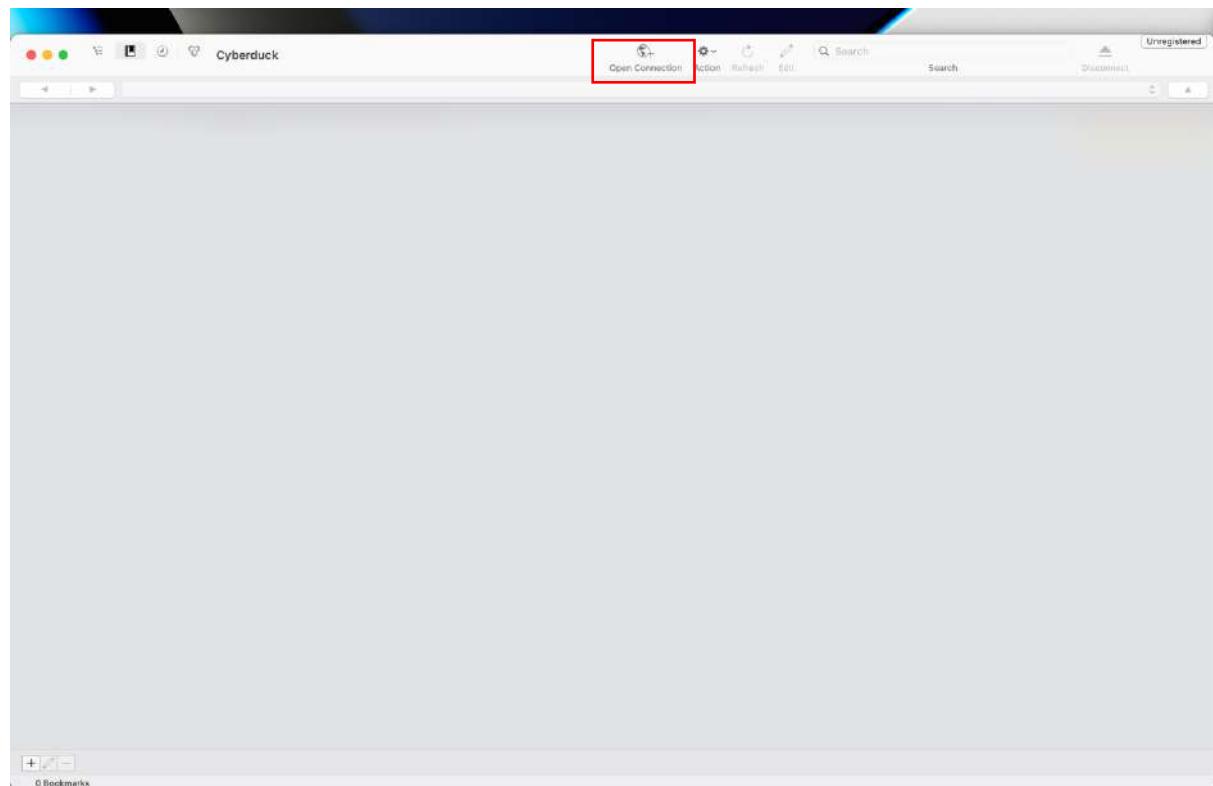
Below the table, the details for the selected instance (i-0f44d275b981ea95a) are shown. The 'Details' tab is active, displaying the following information:

- Public IPv4 address:** 44.217.21.102 [open address]
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-80-128.ec2.internal

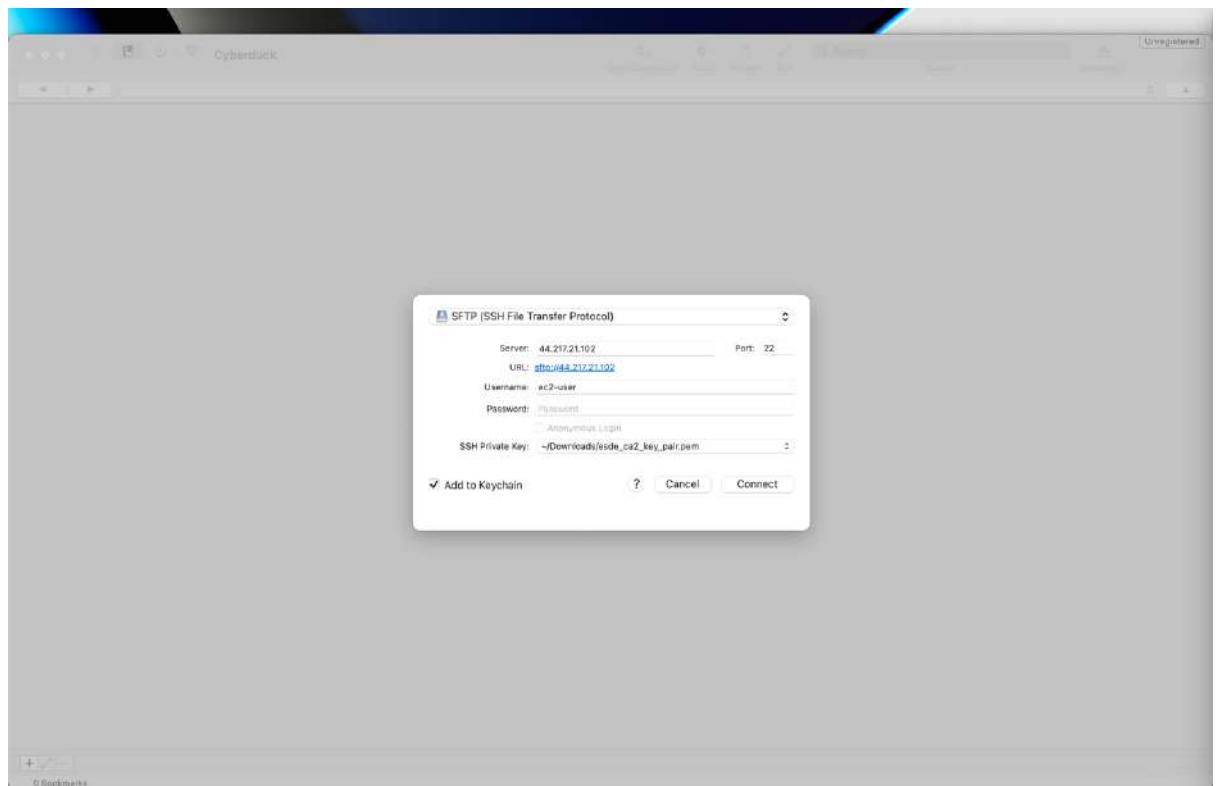
- Copy the public IPv4 address as it is needed for connecting to the instance.
- Download Cyberduck and open it.



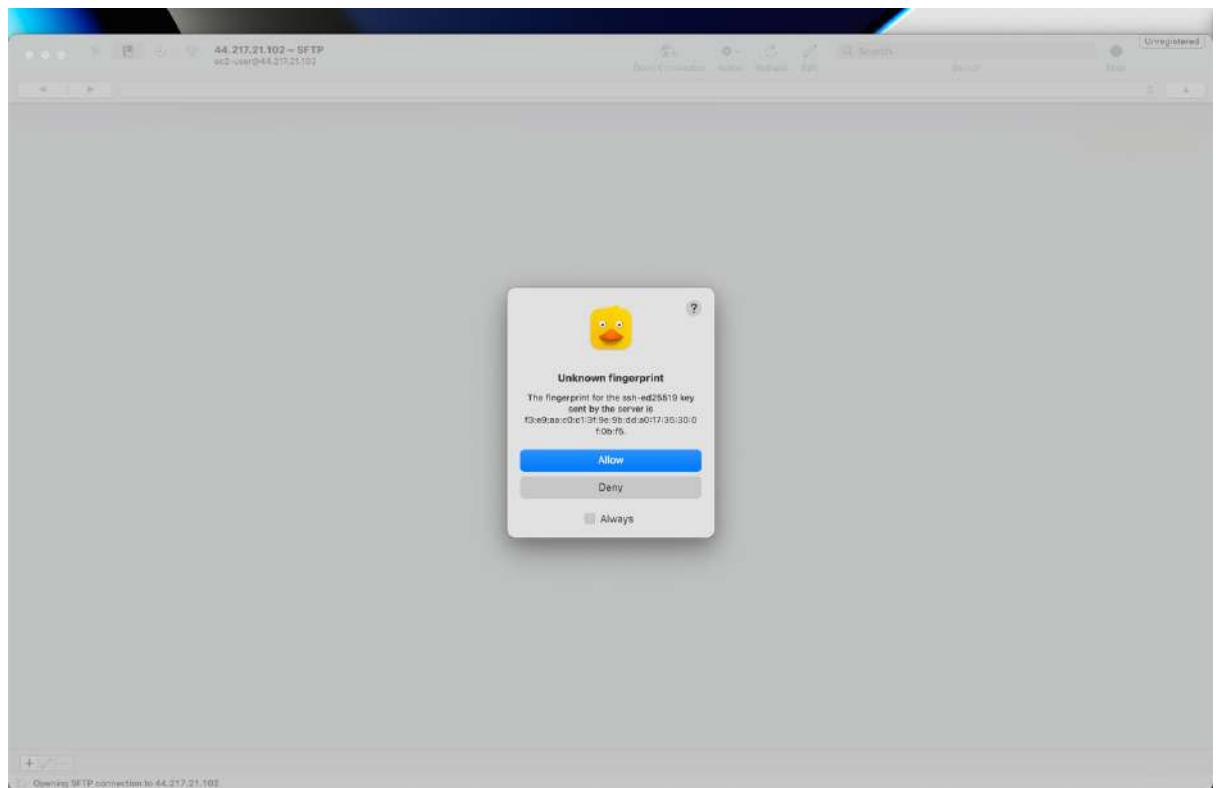
- Click on Open Connection to connect to the instance.



- 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.

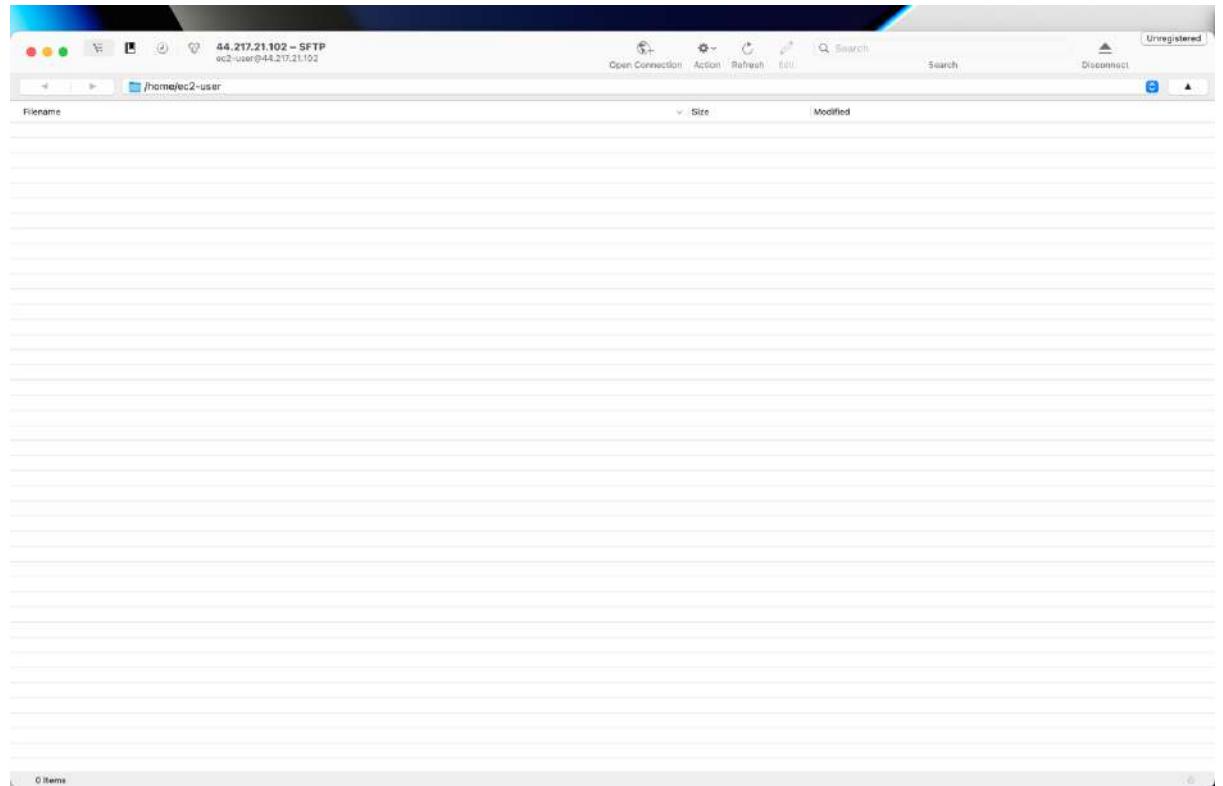


- Choose Allow when Cyberduck prompt for Unknown fingerprint.

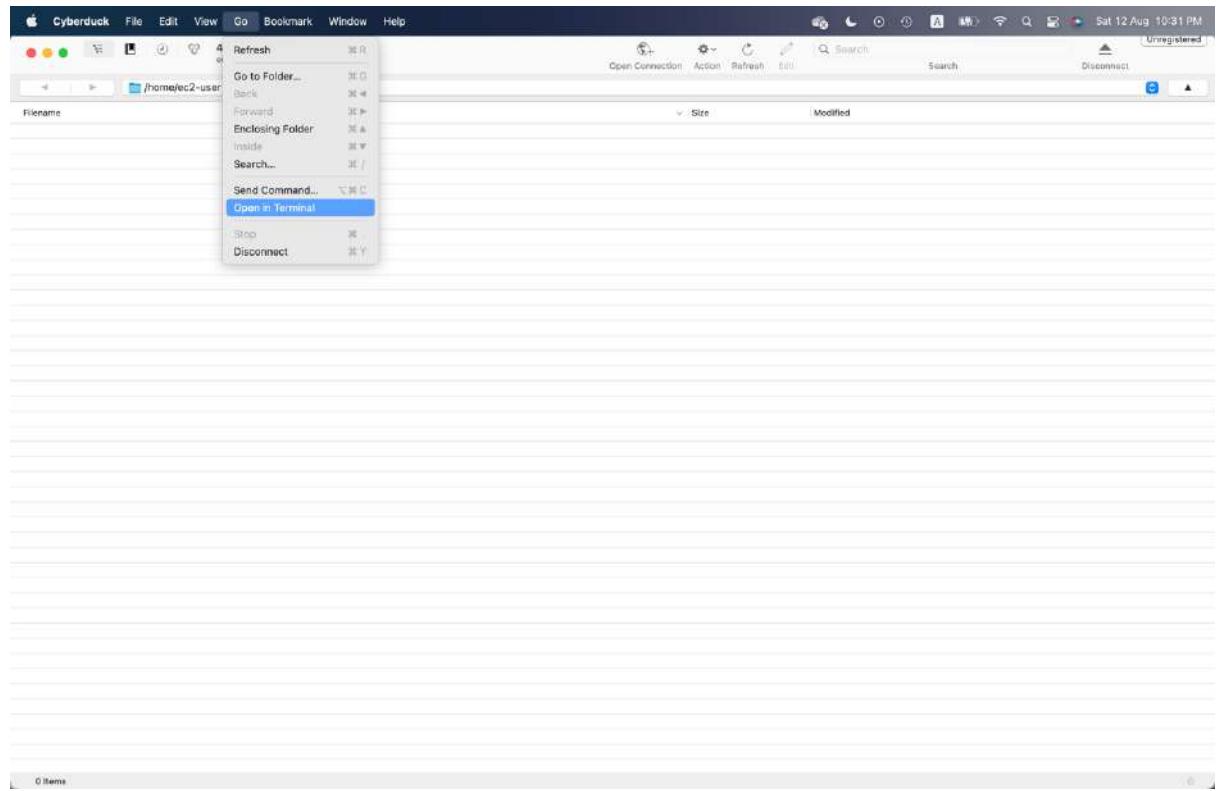


3. Installing NodeJS on EC2 Instance

- 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 -l + the key pair + ec2-user@ + the public IPv4 address**.

```
fan@ec2-user:~$ ssh -i ~/Downloads/esde_ca2_key_pair.pem ec2-user@44.217.21.102 -216x65
Last login: Sat Aug 12 22:37:06 on ttys000
ash -t -l "/Users/fan/Downloads/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd '/home/ec2-user' && exec '$SHELL' -i"
fan@40Fans-MacBook-Pro ~ % ssh -t -l "/Users/fan/Downloads/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd '/home/ec2-user' && exec '$SHELL' -i"
Permissions e644 for '/Users/fan/Downloads/esde_ca2_key_pair.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/Users/fan/Downloads/esde_ca2_key_pair.pem": bad permissions
ash -t -l "/Users/fan/Downloads/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd '/home/ec2-user' && exec '$SHELL' -i"
fan@40Fans-MacBook-Pro ~ % chmod 400 /Users/fan/Downloads/esde_ca2_key_pair.pem
fan@40Fans-MacBook-Pro ~ % ssh -i /Users/fan/Downloads/esde_ca2_key_pair.pem ec2-user@44.217.21.102
.
.
.
Amazon Linux 2023
.
.
.
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user:~]$
```

- ## - Run

```
sudo yum -y install curl
```

```
curl -sL https://rpm.nodesource.com/setup_16.x | sudo
```

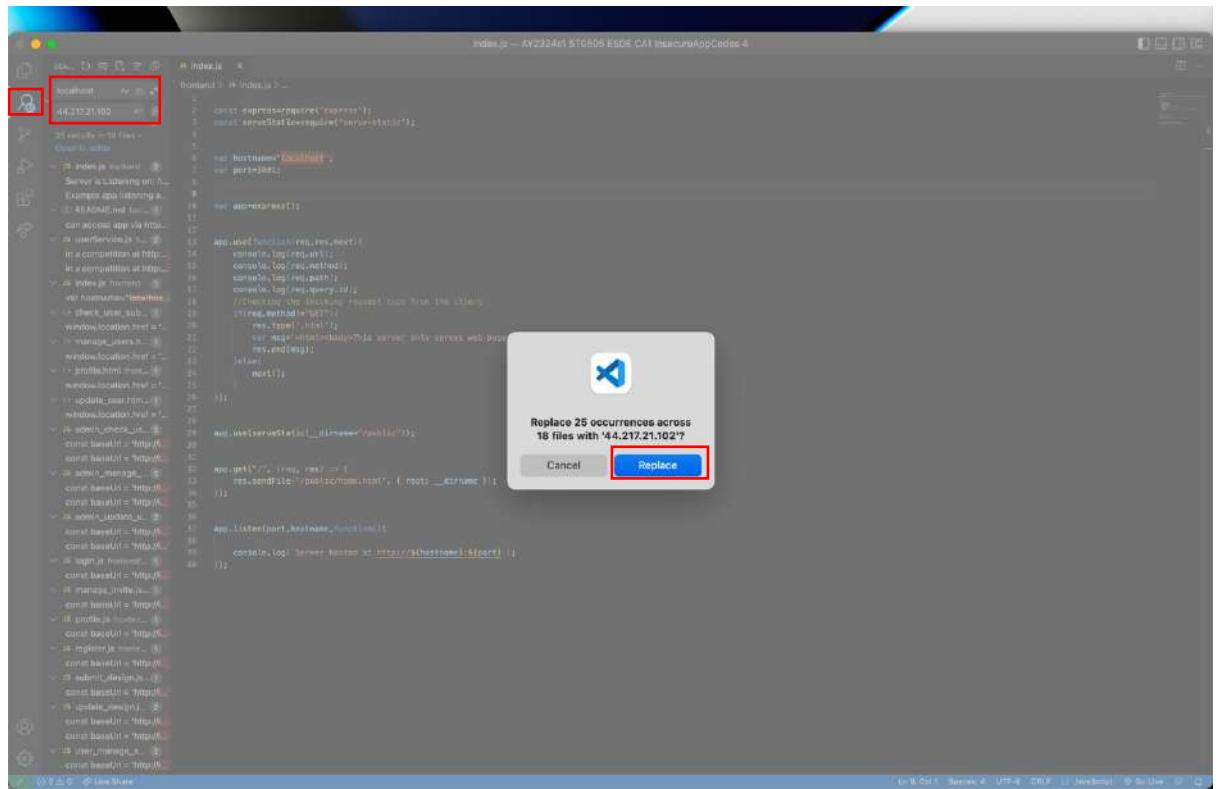
bash -

```
sudo yum install -y nodejs
```

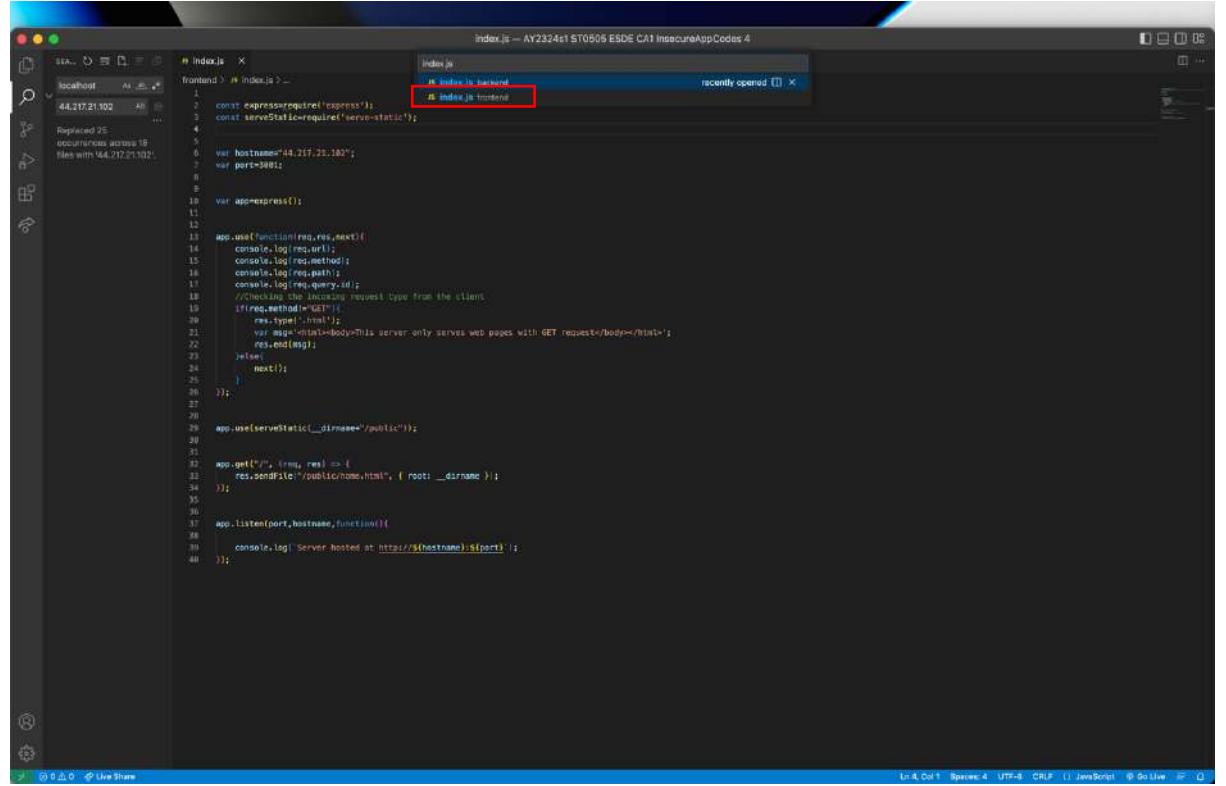
```
sudo yum -y install gcc-c++ make node –version.
```

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.

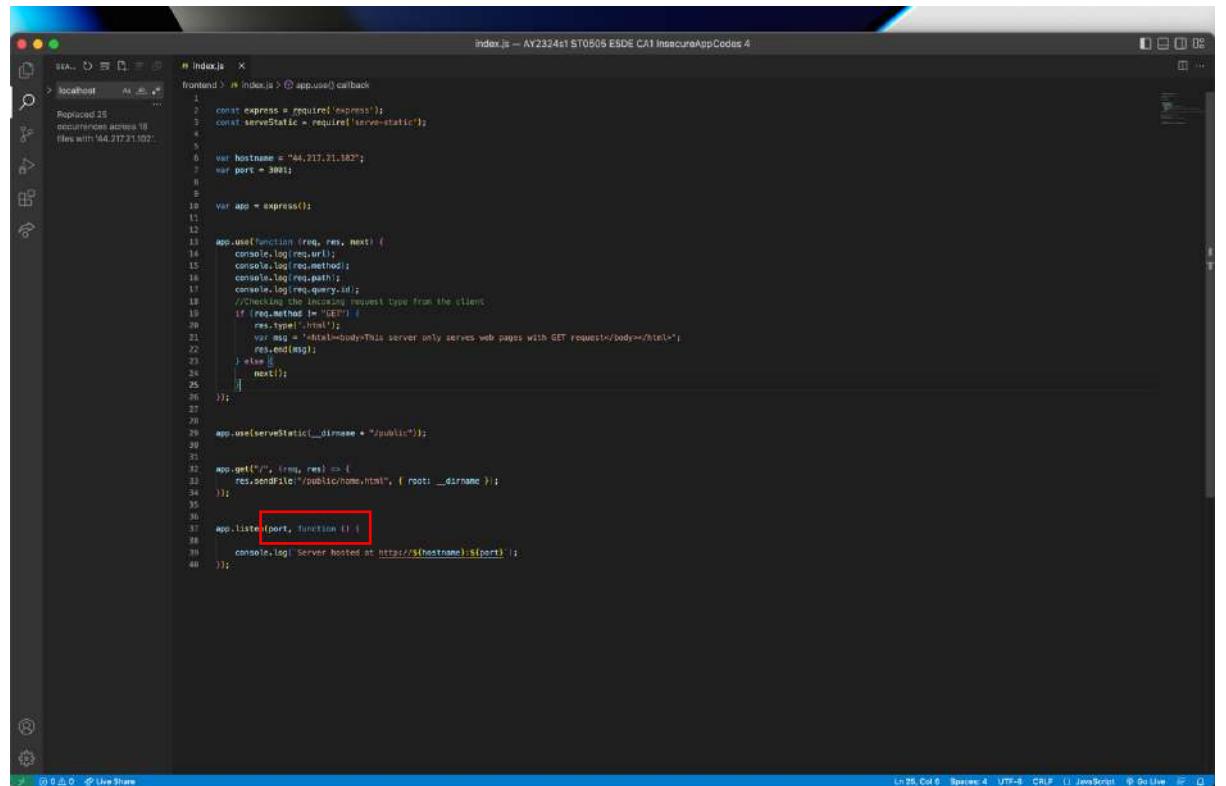


```
index.js - AY2324st ST0505 ESDE CA1 InsecureAppCodes 4
frontend > index.js > ...
localhost 44.217.21.102 All ...
Recently opened: index.js
recently opened: index.js frontend

1 const express = require('express');
2 const serveStatic = require('serve-static');
3
4 var hostname = "44.217.21.102";
5 var port = 3001;
6
7 var app = express();
8
9 app.use(function(req, res, next) {
10   console.log(req.url);
11   console.log(req.method);
12   console.log(req.path);
13   console.log(req.query.id);
14   //Checking the incoming request type from the client
15   if(req.method === "GET") {
16     res.type("html");
17     var msg = "<html><body>This server only serves web pages with GET requests</body></html>";
18     res.end(msg);
19   } else {
20     next();
21   }
22 }
23
24 app.use(serveStatic(__dirname + '/public'));
25
26 app.get('/', (req, res) => {
27   res.sendFile('/public/home.html', { root: __dirname });
28 })
29
30 app.listen(port, hostname, function() {
31   console.log(`Server hosted at https://${hostname}:${port}`);
32 });

Line 40 Col 6 Spacing: 4 UTF-8 CR/LF: ( ) JavaScript Go Live
```

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

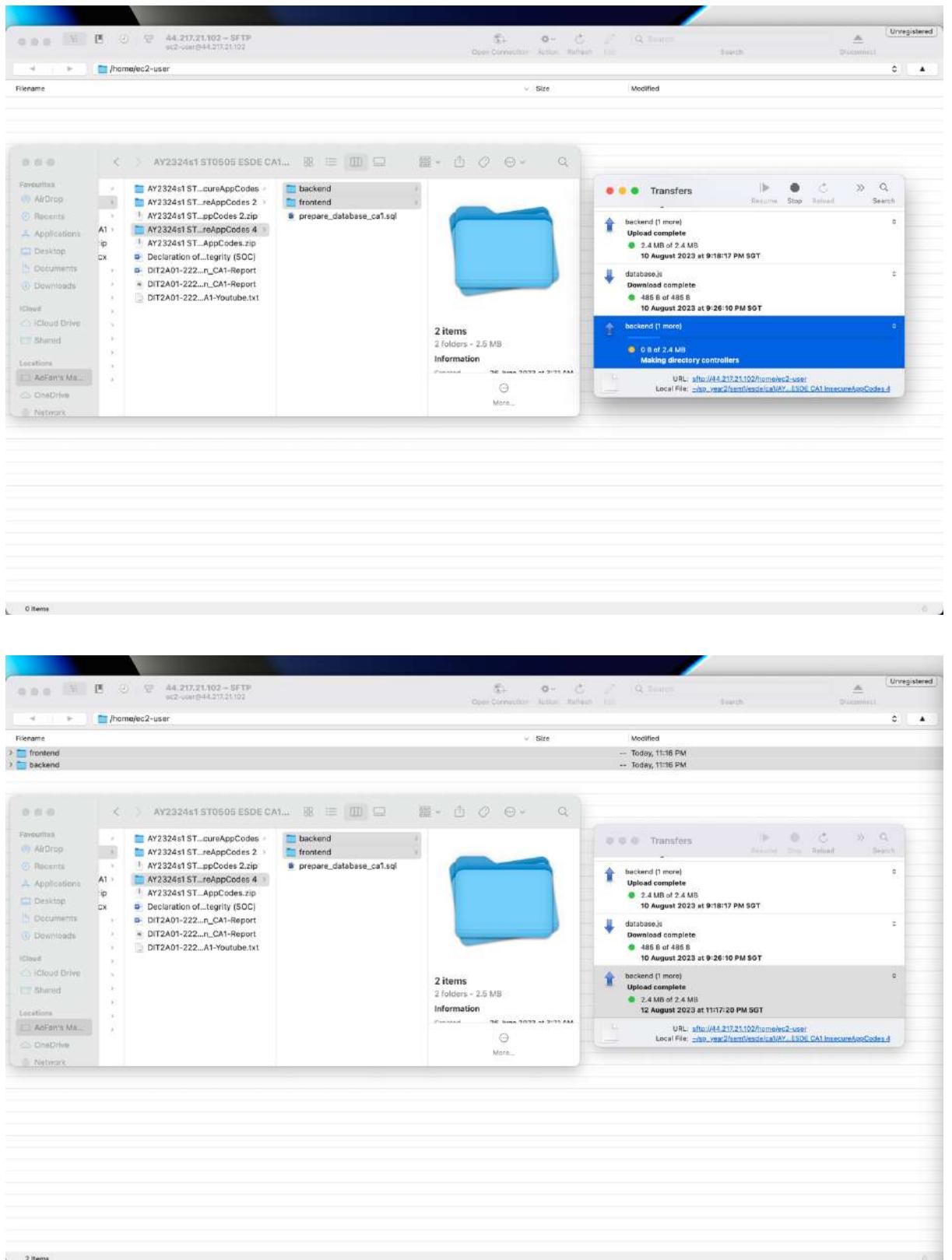


```
index.js - AY2324st ST0505 ESDE CA1 InsecureAppCodes 4
frontend > index.js > @ app.use callback
localhost 44.217.21.102 All ...
Recently opened: index.js
recently opened: index.js frontend

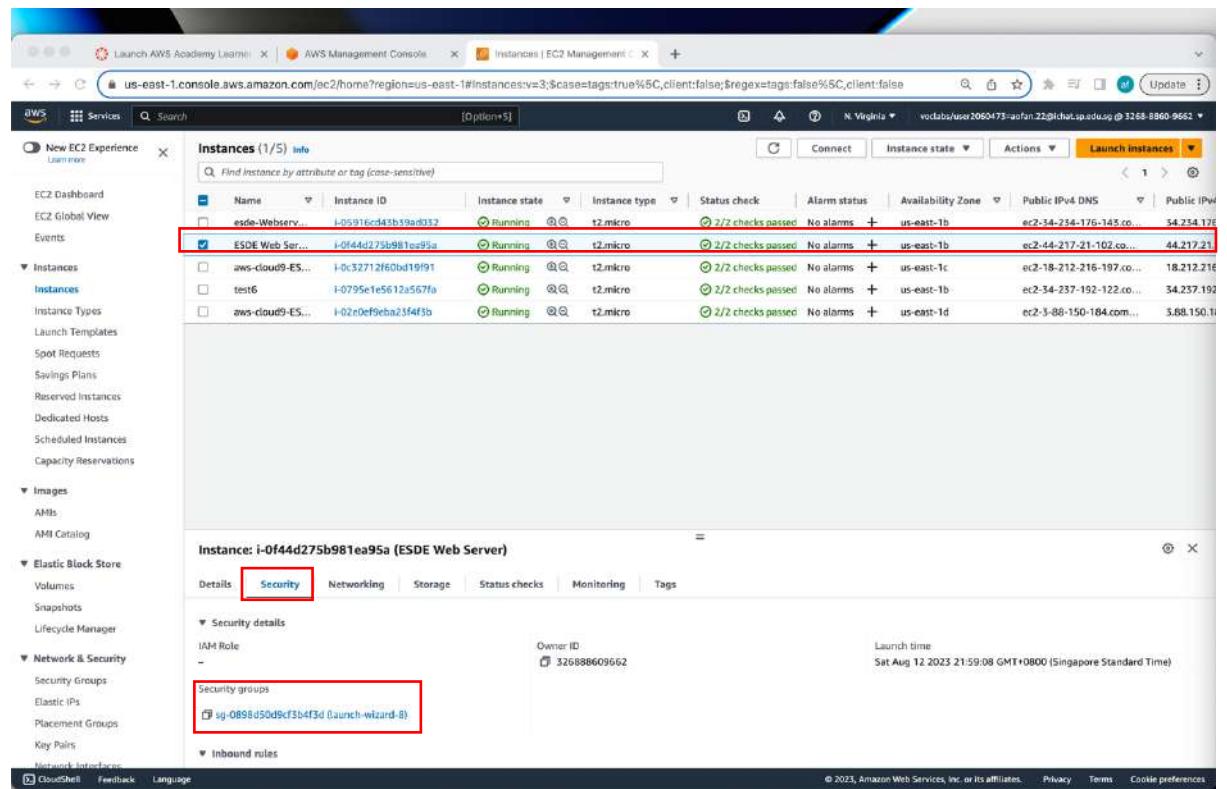
1 const express = require('express');
2 const serveStatic = require('serve-static');
3
4 var hostname = "44.217.21.102";
5 var port = 3001;
6
7 var app = express();
8
9 app.use(function(req, res, next) {
10   console.log(req.url);
11   console.log(req.method);
12   console.log(req.path);
13   console.log(req.query.id);
14   //Checking the incoming request type from the client
15   if (req.method === "GET") {
16     res.type("html");
17     var msg = "<html><body>This server only serves web pages with GET requests</body></html>";
18     res.end(msg);
19   } else {
20     next();
21   }
22 }
23
24 app.use(serveStatic(__dirname + '/public'));
25
26 app.get('/', (req, res) => {
27   res.sendFile('/public/home.html', { root: __dirname });
28 })
29
30 app.listen(port, function () {
31   console.log(`Server hosted at https://${hostname}:${port}`);
32 });

Line 40 Col 6 Spacing: 4 UTF-8 CR/LF: ( ) JavaScript Go Live
```

- 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.

Details

Security group name	sg-0898d50d9cf3b4f3d	Description	vpc-075cc676504d92f8e
Owner	326888609662	Inbound rules count	1 Permission entry
		Outbound rules count	1 Permission entry

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-03bcd5494d7fad56f	IPv4	SSH	TCP	22	0.0.0.0/0

Manage tags | Edit inbound rules

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

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

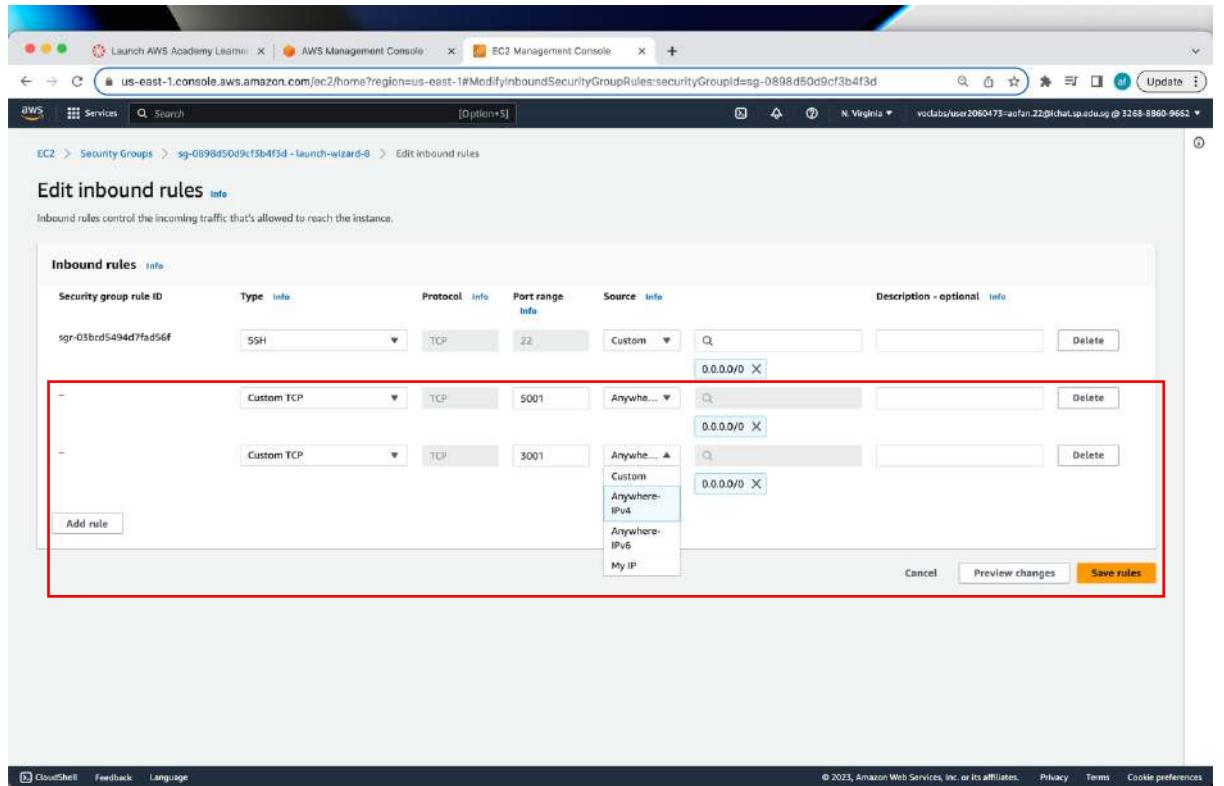
Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03bcd5494d7fad56f	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	0	Custom	0.0.0.0/0
-	Custom TCP	TCP	0	Custom	0.0.0.0/0

Add rule

Cancel | Preview changes | Save rules

- 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.

```

fan@ec2-user:~$ npm audit
Last Login: Sat Aug 12 23:32:27 on ttys000
ssh -t -i "/Users/fan/.ssh/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd ~/Home/ec2-user && exec '$SHELL -l'" fan@Oasis-MacBook-Pro: ~ fan$ ssh -t -i "/Users/fan/.ssh/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd ~/Home/ec2-user && exec '$SHELL -l'" fan@ec2-user:~$ cd frontend
[ec2-user@ip-172-31-88-128 frontend]$ npm i
[ec2-user@ip-172-31-88-128 frontend]$ npm i
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies
npm WARN deprecated axios@0.19.2: Critical security vulnerability fixed in v0.21.1. For more information, see https://github.com/axios/axios/pull/3418
added 274 packages, and audited 276 packages in 1s
added 274 packages, and audited 276 packages in 1s

15 vulnerabilities (5 moderate, 9 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

```



```

fan@ec2-user:~$ npm audit
Last login: Sat Aug 12 23:32:34 on ttys001
ssh -t -i "/Users/fan/.ssh/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd ~/Home/ec2-user && exec '$SHELL -l'" fan@Oasis-MacBook-Pro: ~ fan$ ssh -t -i "/Users/fan/.ssh/esde_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd ~/Home/ec2-user && exec '$SHELL -l'" fan@ec2-user:~$ cd backend
[ec2-user@ip-172-31-88-128 backend]$ npm i
[ec2-user@ip-172-31-88-128 backend]$ npm i
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies
npm WARN deprecated axios@0.19.2: Critical security vulnerability fixed in v0.21.1. For more information, see https://github.com/axios/axios/pull/3418
added 482 packages, and audited 483 packages in 2s
16 packages are looking for funding
  run 'npm fund' for details

```

- Run **npm start** for both terminals

```

fan@ec2-user:~$ npm audit
To address issues that do not require attention, run:
  npm audit fix
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
npm notice New major version of npm available! 8.19.2 => 9.0.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.0.1
npm notice Run `npm install -g npm@9.0.1` to update!
[ec2-user@ip-172-31-88-128 frontend]$ npm start
> firstfrontand1.0.0 start
> nodemon index.js
[nodemon] 2.1.9
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching directory: ./
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node index.js'
Server hosted at http://44.217.21.102:3001

```

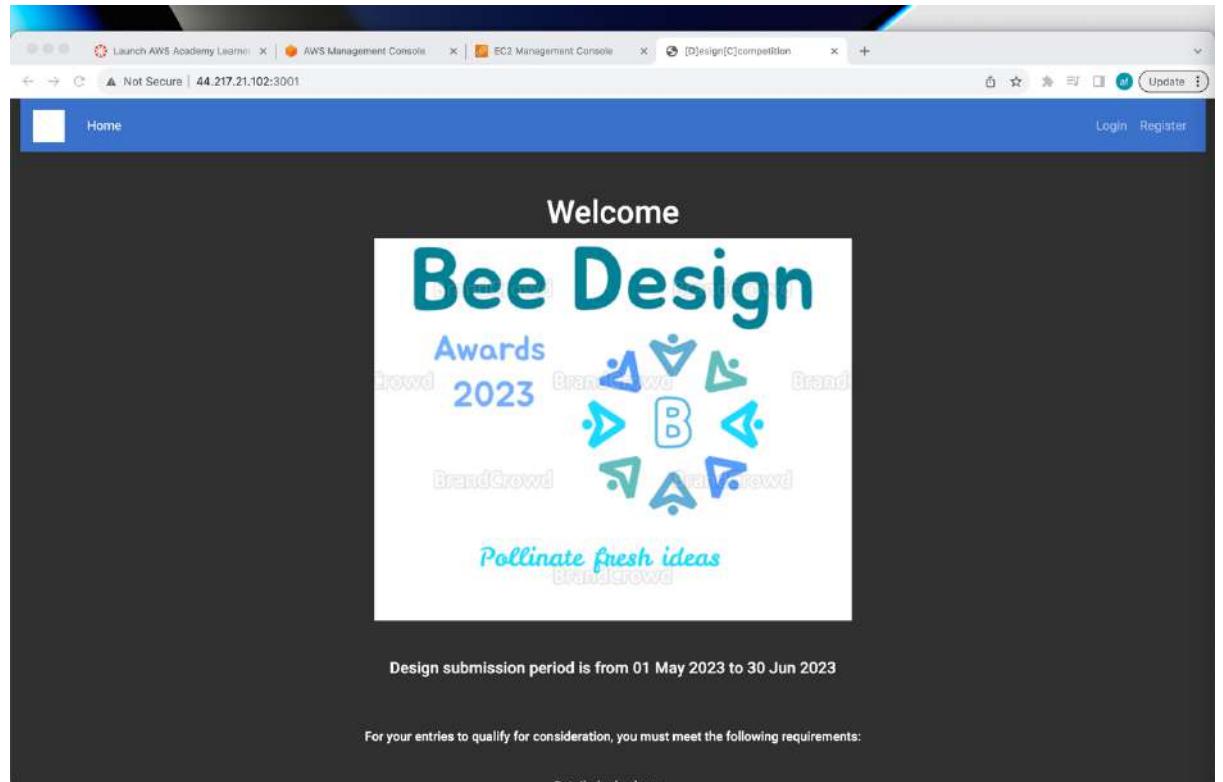


```

fan@ec2-user:~$ npm audit
To address issues that do not require attention, run:
  npm audit fix
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
npm notice New major version of npm available! 8.19.2 => 9.0.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.0.1
npm notice Run `npm install -g npm@9.0.1` to update!
[ec2-user@ip-172-31-88-128 backend]$ npm start
> xyz1.0.0 start
> nodemon index.js
[nodemon] 2.1.9
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching directory: ./
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node index.js'
Example app listening at http://44.217.21.102:5001

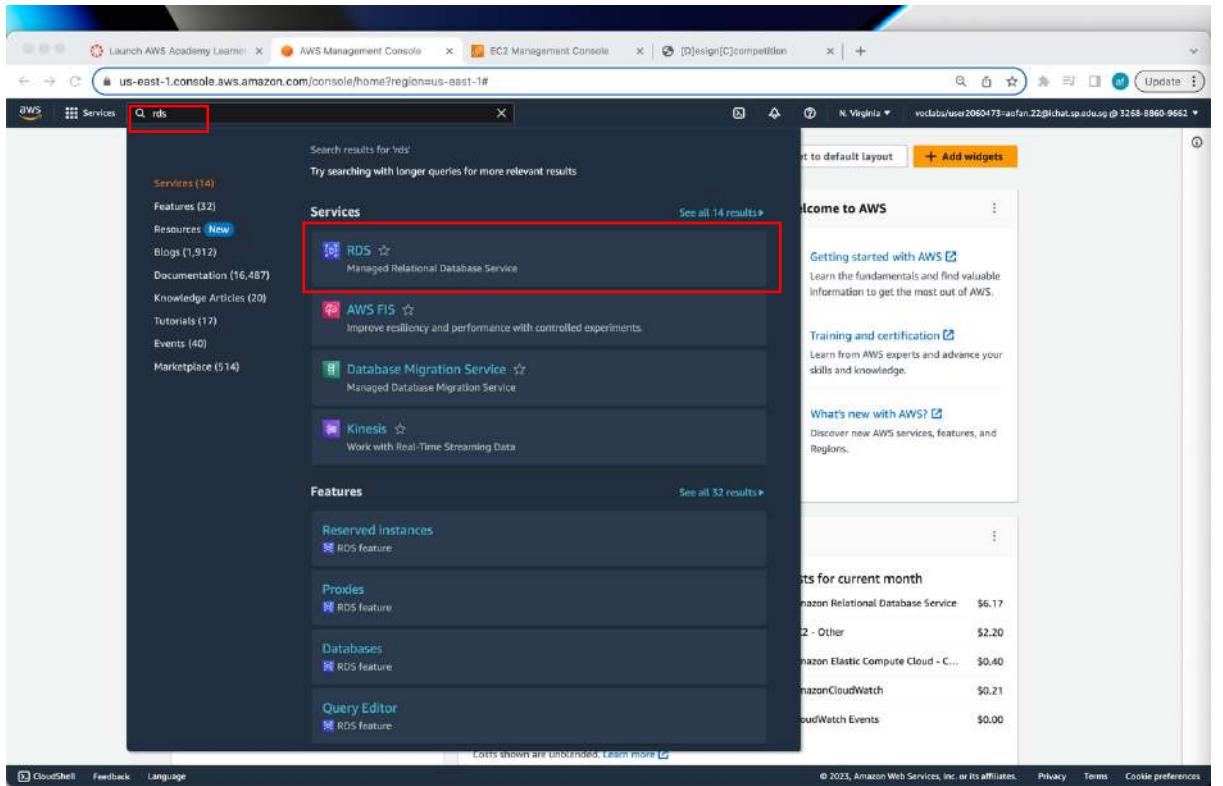
```

- 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.

The screenshot shows the Amazon RDS Dashboard. At the top, there is a banner about Aurora I/O-Optimized. Below it, a callout box highlights the 'Create database' button. The main area displays various RDS resources like DB Instances, DB Clusters, and Snapshots. On the right, there are 'Recommended for you' links and an 'Additional information' section.

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

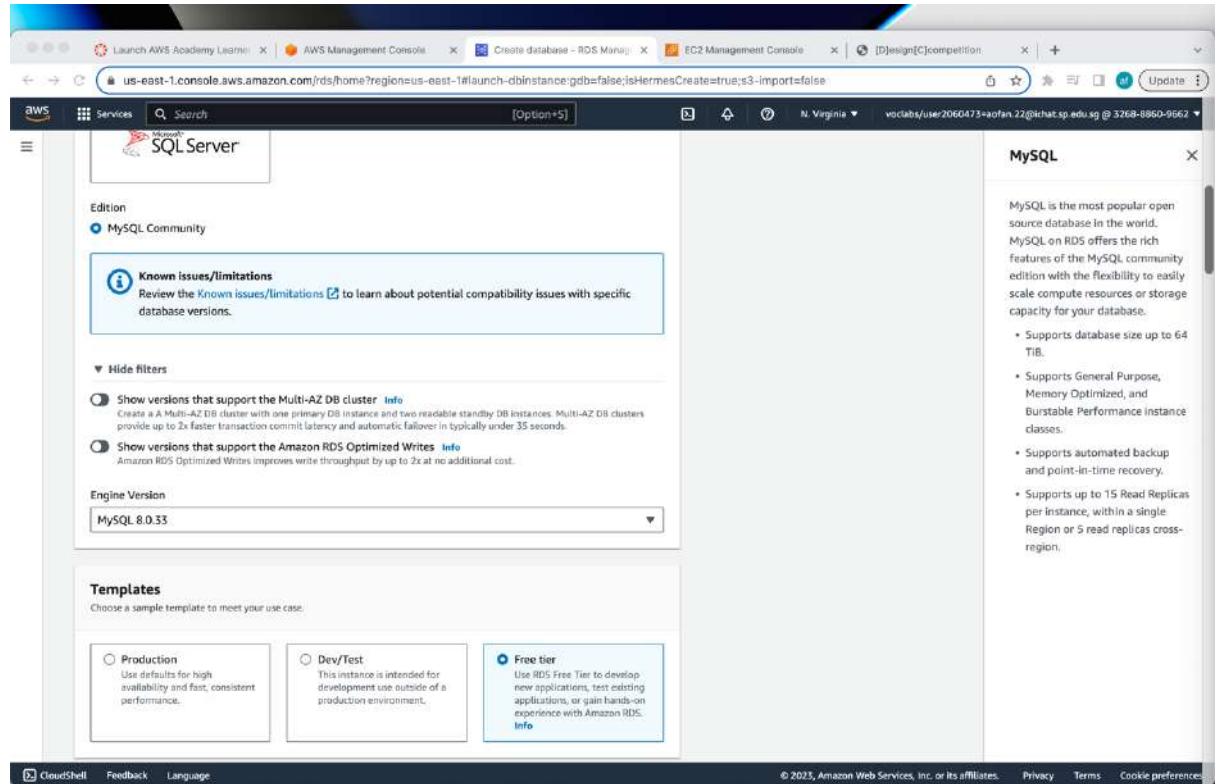
The screenshot shows the 'Create database' wizard. Under 'Choose a database creation method', 'Standard create' is selected. Under 'Engine options', 'MySQL' is selected. To the right, there is a detailed description of MySQL and a list of its features.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

- 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.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

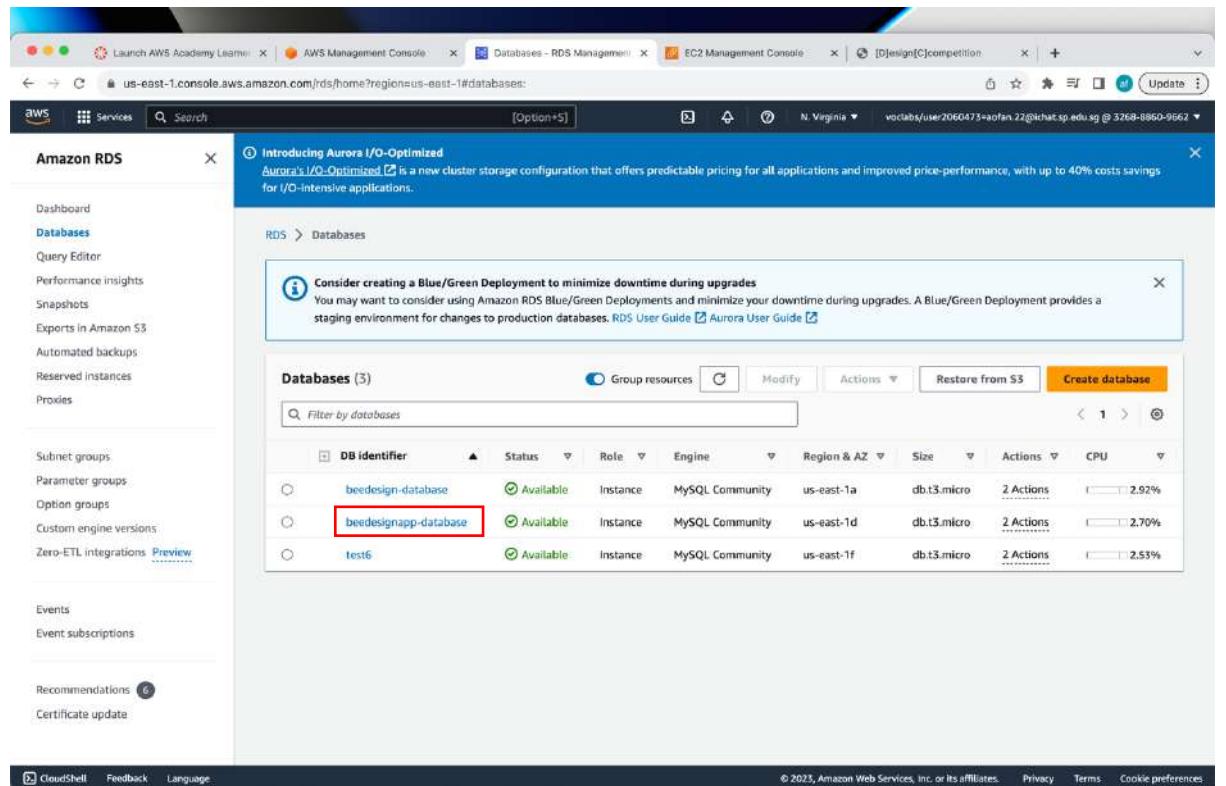
- Under Connectivity, select Yes for Public access.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

- 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.



The screenshot shows the AWS RDS Management Console interface. On the left, there's a sidebar with various options like Dashboard, Databases (which is selected), Query Editor, etc. The main area is titled 'Databases (3)' and lists three databases:

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU
beedesign-database	Available	Instance	MySQL Community	us-east-1a	db.t3.micro	2 Actions	2.92%
beedesignapp-database	Available	Instance	MySQL Community	us-east-1d	db.t3.micro	2 Actions	2.70%
test6	Available	Instance	MySQL Community	us-east-1f	db.t3.micro	2 Actions	2.55%

A modal window titled 'Consider creating a Blue/Green Deployment to minimize downtime during upgrades' is open, providing information about RDS Blue/Green Deployments.

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

Summary

DB identifier beedesignapp-database	CPU 2.70%	Status Available	Class db.t3.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1d

Connectivity & security

Endpoint & port	Networking	Security
Endpoint beedesignapp-database.cn0xrwhwuqj.us-east-1.rds.amazonaws.com	Availability Zone us-east-1d	VPC security groups default (sg-0c19874065ab62af2) Active
Port 3306	VPC vpc-075cc676504d92f8e	Publicly accessible Yes
	Subnet group default-vpc-075cc676504d92f8e	Certificate authority Info rds-ca-2019
	Subnets subnet-0a3fb85041f0b6743	Certificate authority date

- Click on Inbound rules followed by Edit inbound rules

Security Groups (1/1) [Info](#)

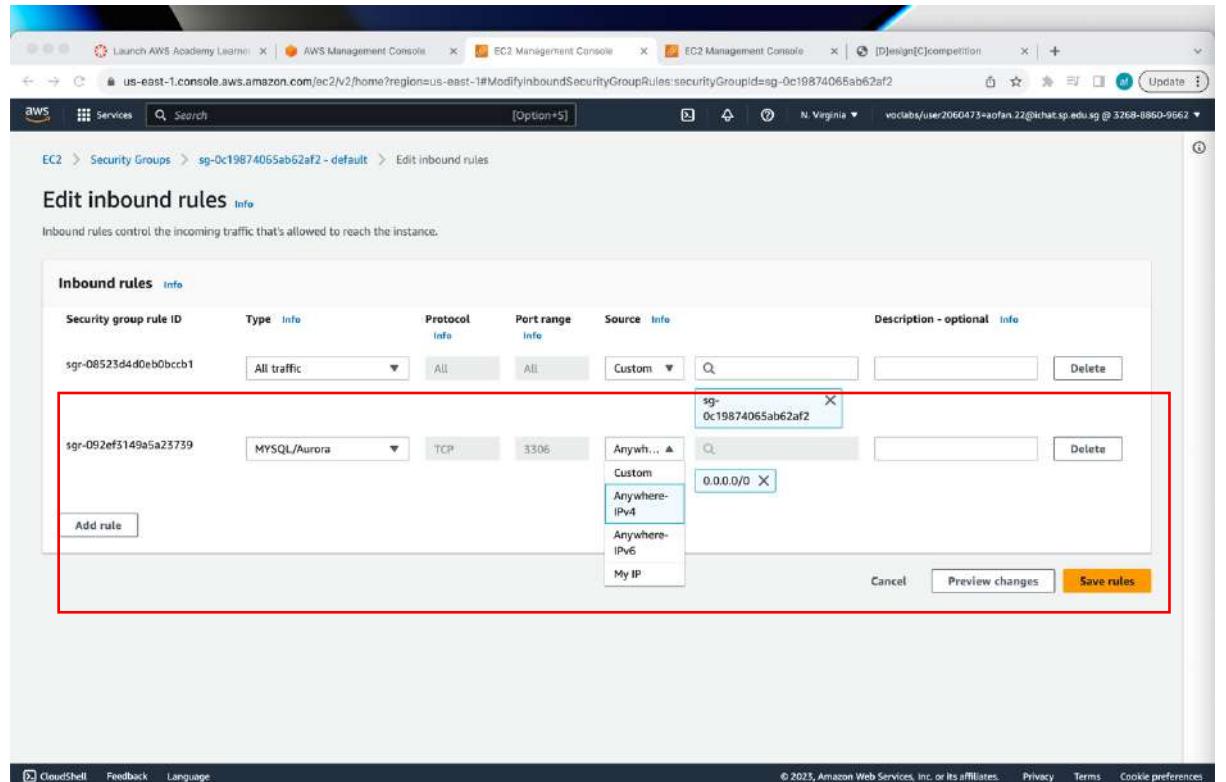
Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0c19874065ab62af2	default	vpc-075cc676504d92f8e	default VPC security gr...	32688609662

sg-0c19874065ab62af2 - default

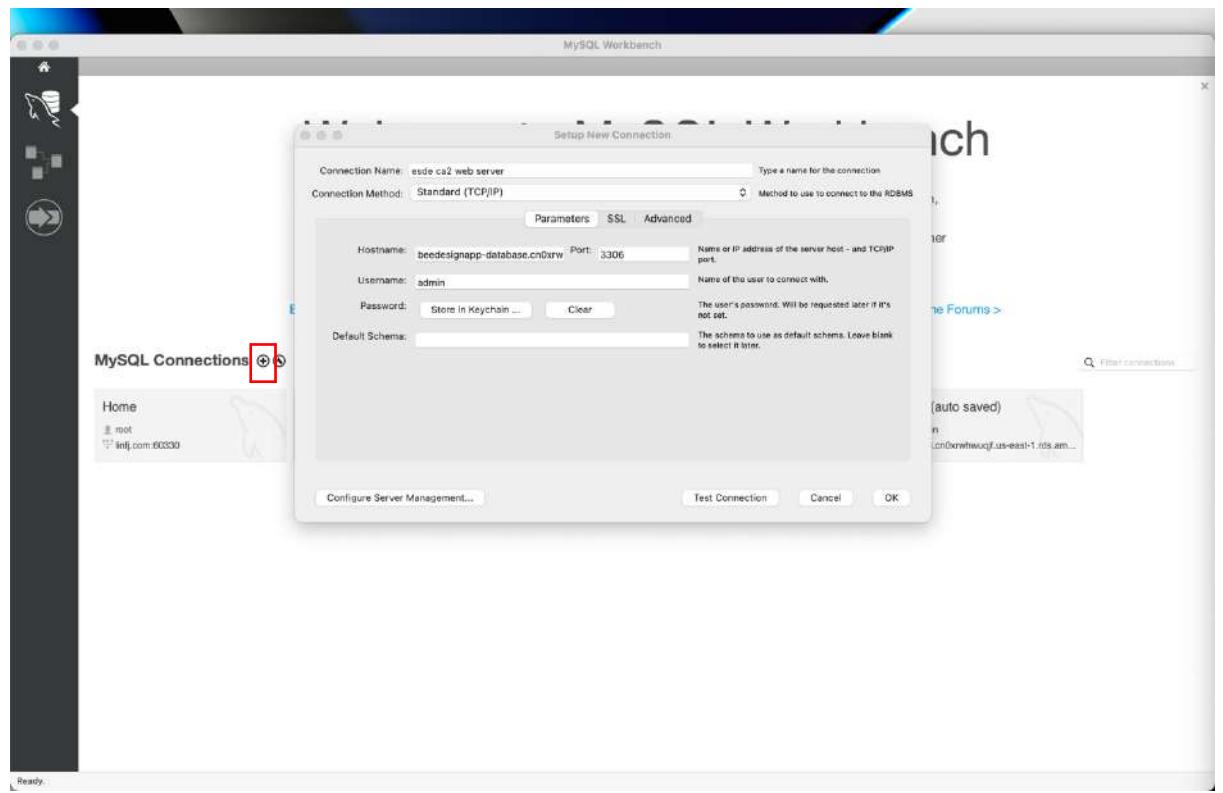
Inbound rules (2)

[Edit inbound rules](#)

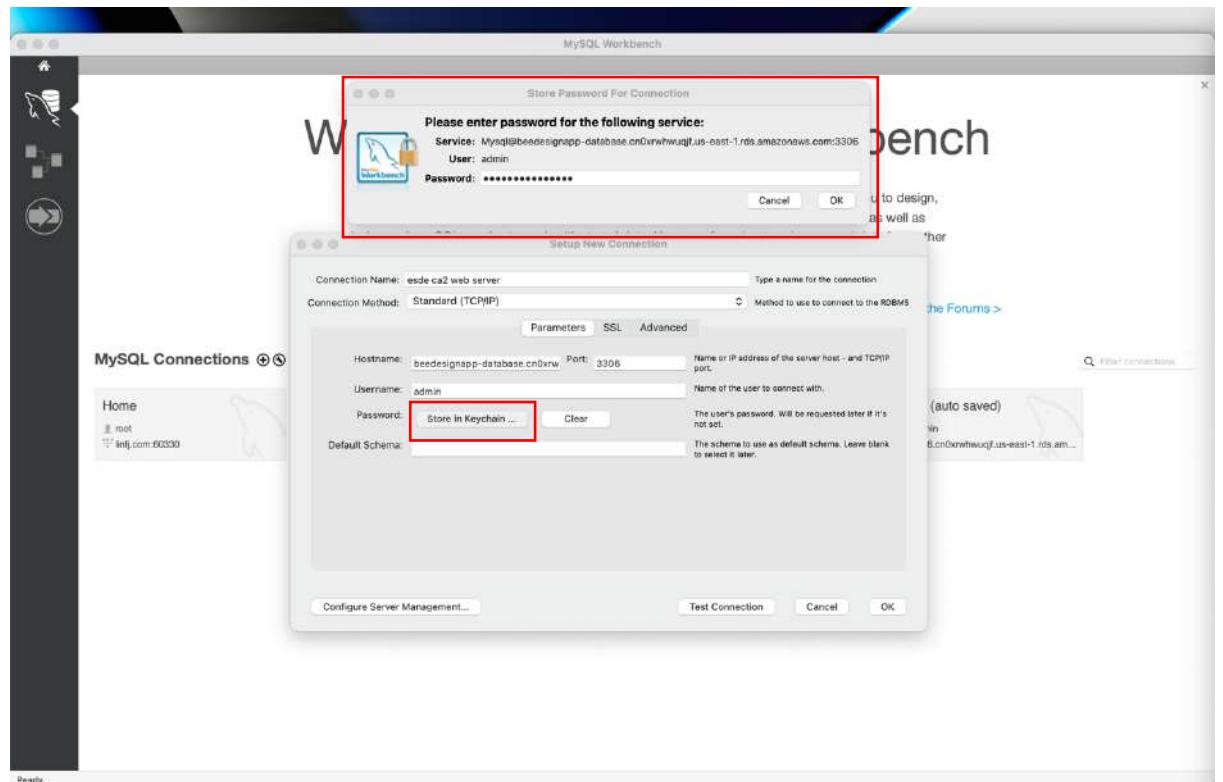
- Click Add rule, select MYSQL/Aurora under Type, select Anywhere-IPv4 under Source and click Save rules. (If you have the inbound rule for MySQL you can skip this step)



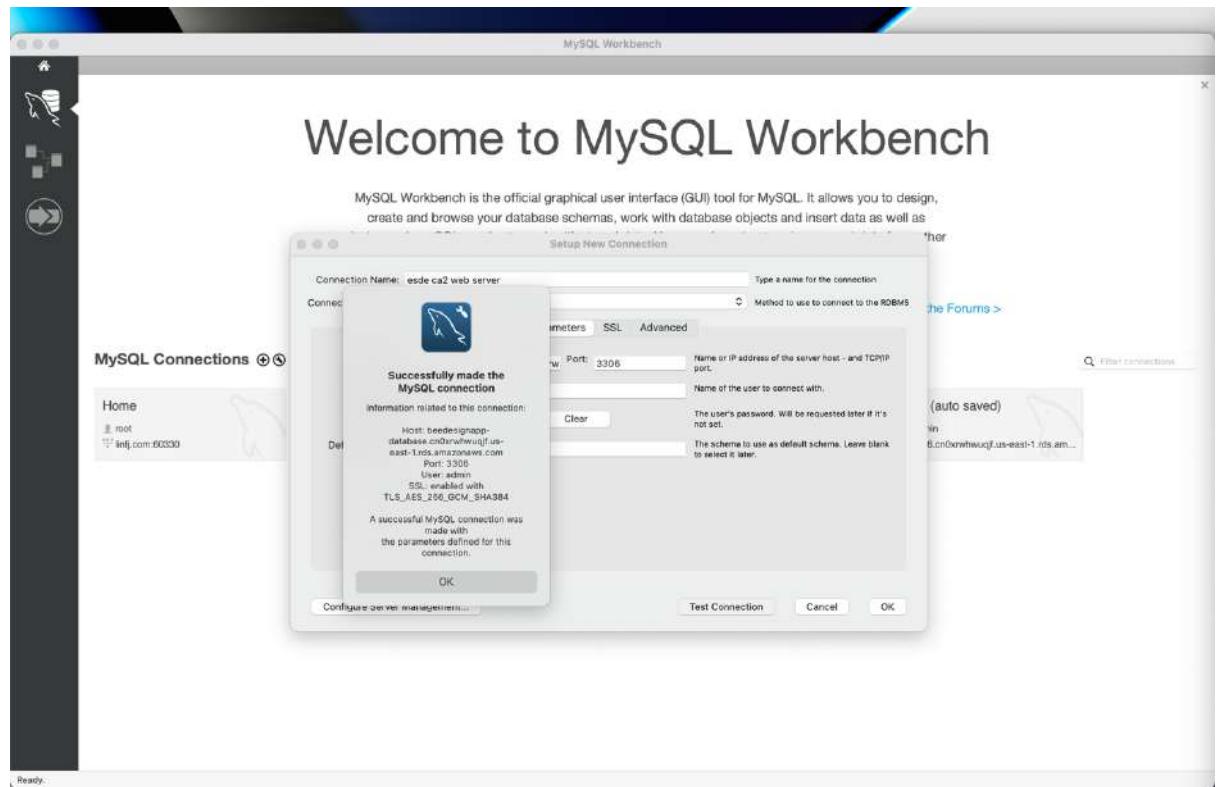
- Open MySQLWorkbench, click on the plus icon to setup a new connection. Enter esde ca2 web server in Connection Name:, paste the Endpoint of the RDS database in Hostname:, enter the Master username (admin) in Username::



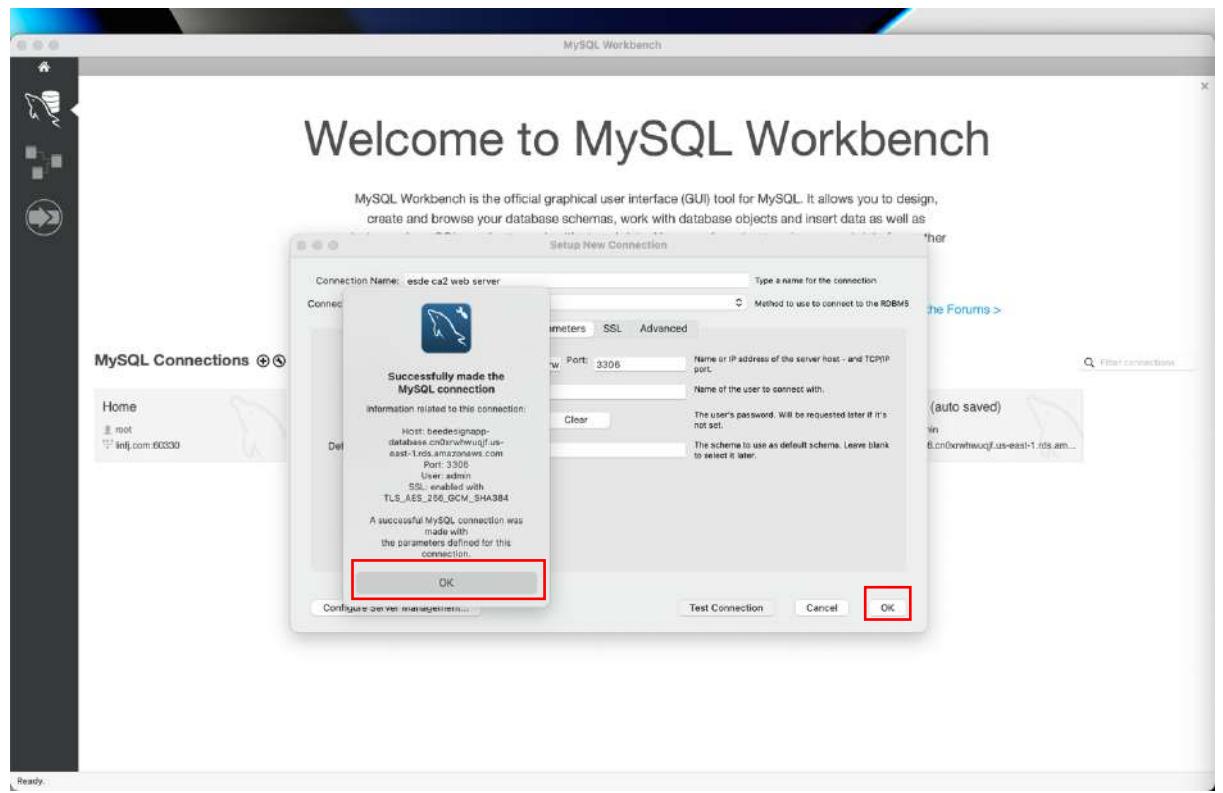
- Click on Store in Keychain .. and enter the Master password. (The password set when creating RDS database)



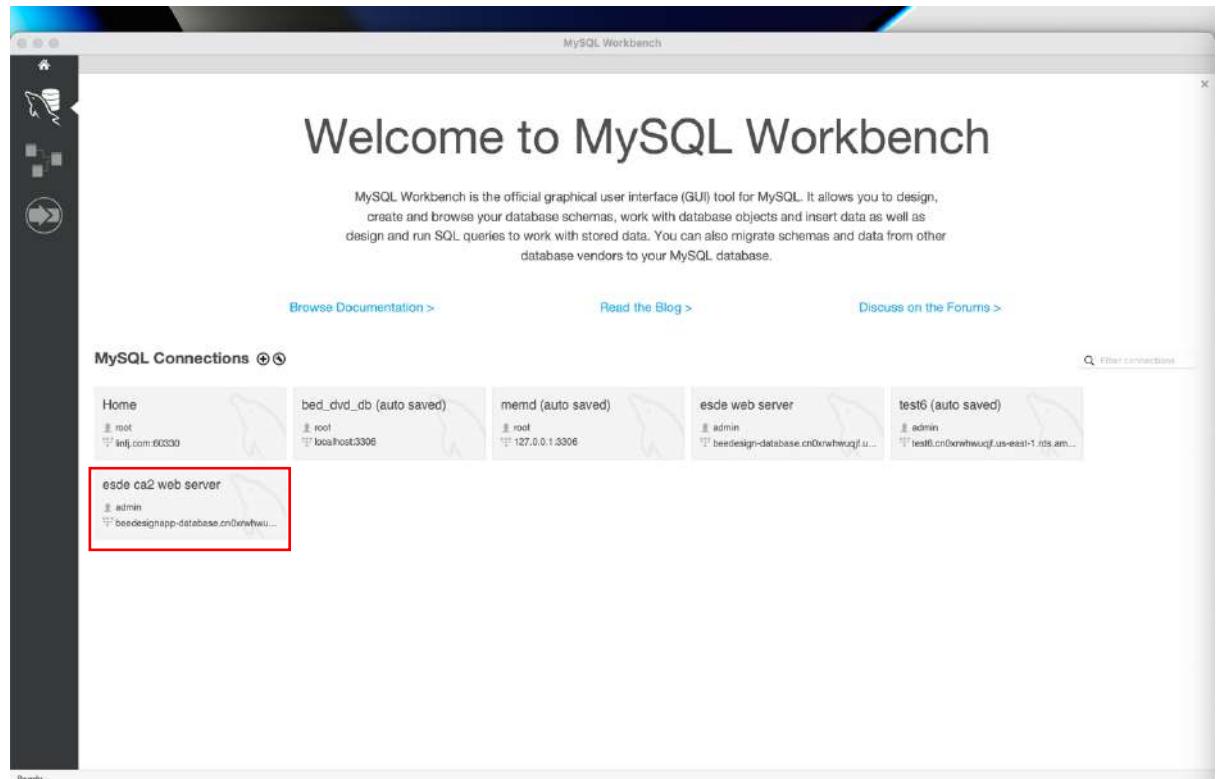
- Click Test Connection to verified that the connection can be established.



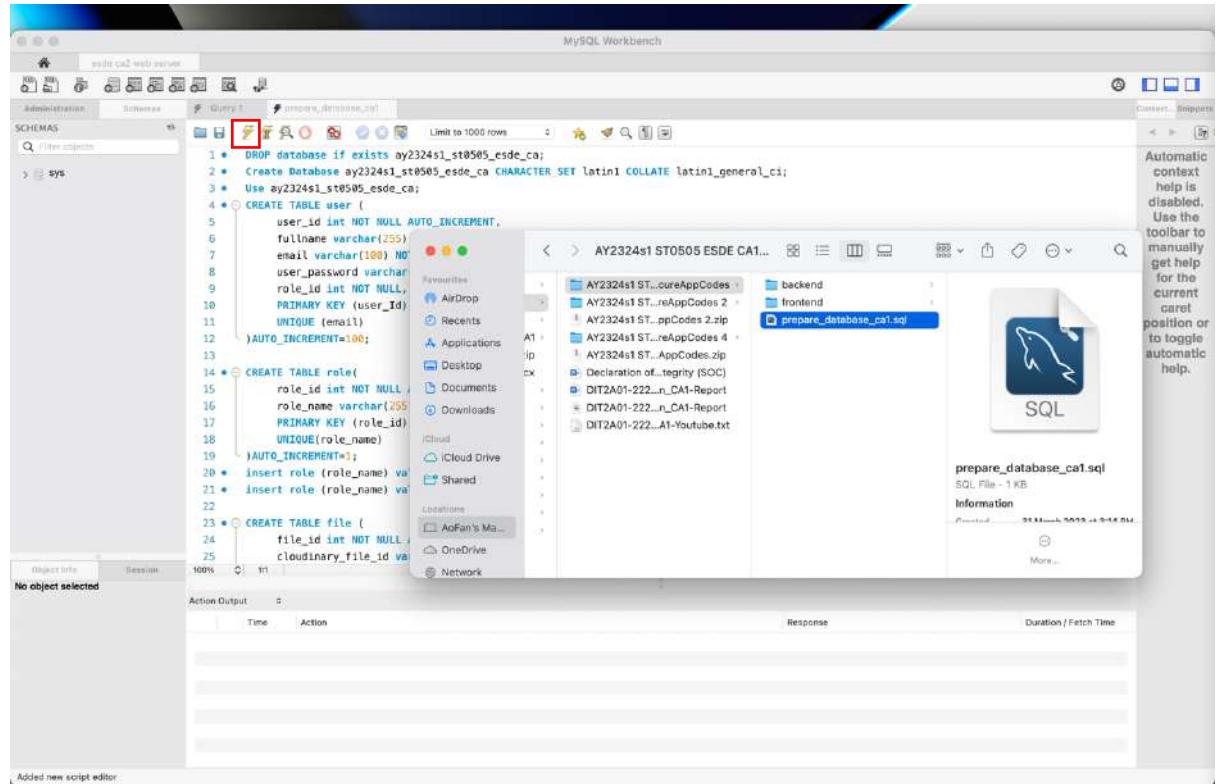
- Click OK to close the Test Connection result and click the OK button to establish the connection.



- 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.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** A red box highlights the "ay2324s1_st0505_esde_ca" schema in the left sidebar.
- Query Editor:** The "Query 1" tab contains the following SQL code:


```

1 • DROP database if exists ay2324s1_st0505_esde_ca;
2 • Create Database ay2324s1_st0505_esde_ca CHARACTER SET latin1 COLLATE latin1_general_ci;
3 • Use ay2324s1_st0505_esde_ca;
4 • CREATE TABLE user (
5     user_id int NOT NULL AUTO_INCREMENT,
6     fullname varchar(255) NOT NULL,
7     email varchar(180) NOT NULL,
8     user_password varchar(255),
9     role_id int NOT NULL,
10    PRIMARY KEY (user_Id),
11    UNIQUE (email)
12 )AUTO_INCREMENT=100;
13
14 • CREATE TABLE role(
15     role_id int NOT NULL AUTO_INCREMENT,
16     role_name varchar(255) NOT NULL,
17     PRIMARY KEY (role_id),
18     UNIQUE(role_name)
19 )AUTO_INCREMENT=1;
20 • insert role (role_name) values ('admin');
21 • insert role (role_name) values ('user');
22
23 • CREATE TABLE file (
24     file_id int NOT NULL AUTO_INCREMENT,
25     cloudinary_file_id varchar(255) NOT NULL,

```
- Action Output:** Shows the execution log with 11 entries, each detailing a query execution time, action, response (number of rows affected), and duration/fetch time.

The screenshot shows the MySQL Workbench interface with the following details:

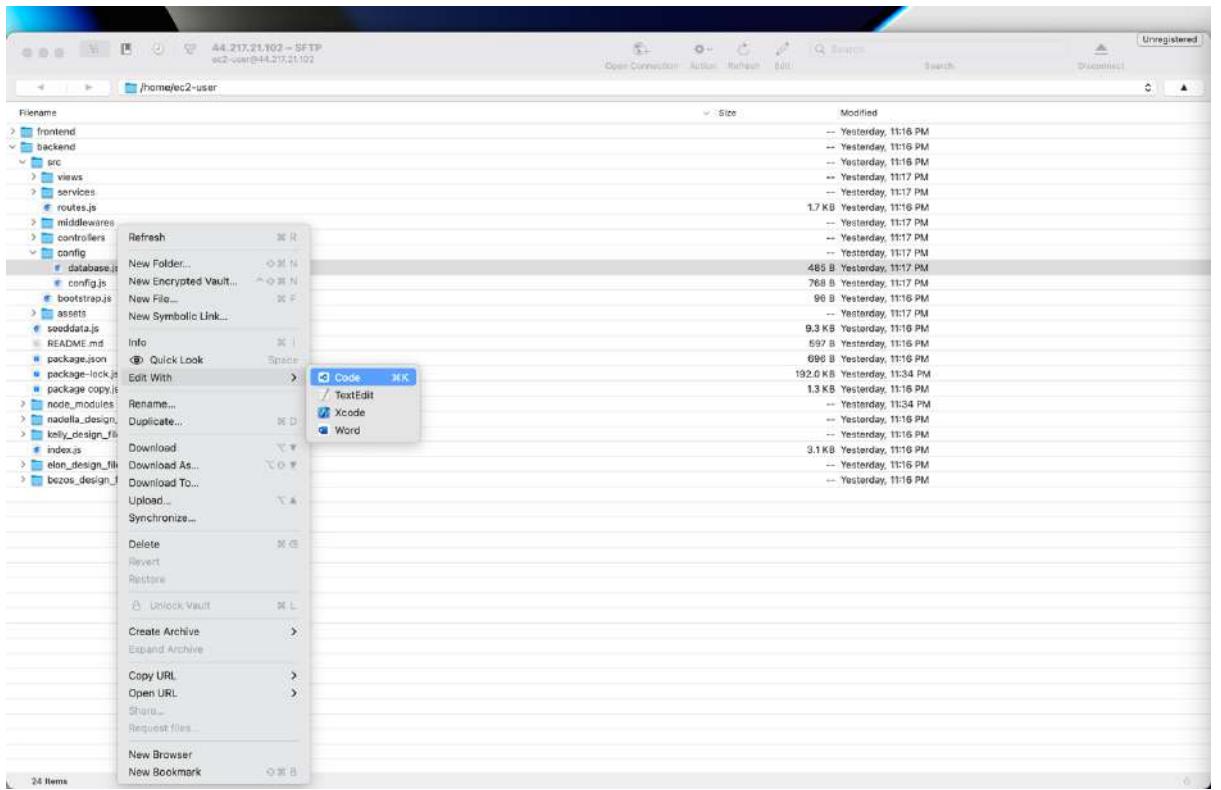
- Schemas:** A red box highlights the "ay2324s1_st0505_esde_ca" schema in the left sidebar.
- Query Editor:** The "Query 1" tab contains the same SQL code as the first screenshot:


```

1 • DROP database if exists ay2324s1_st0505_esde_ca;
2 • Create Database ay2324s1_st0505_esde_ca CHARACTER SET latin1 COLLATE latin1_general_ci;
3 • Use ay2324s1_st0505_esde_ca;
4 • CREATE TABLE user (
5     user_id int NOT NULL AUTO_INCREMENT,
6     fullname varchar(255) NOT NULL,
7     email varchar(180) NOT NULL,
8     user_password varchar(255),
9     role_id int NOT NULL,
10    PRIMARY KEY (user_Id),
11    UNIQUE (email)
12 )AUTO_INCREMENT=100;
13
14 • CREATE TABLE role(
15     role_id int NOT NULL AUTO_INCREMENT,
16     role_name varchar(255) NOT NULL,
17     PRIMARY KEY (role_id),
18     UNIQUE(role_name)
19 )AUTO_INCREMENT=1;
20 • insert role (role_name) values ('admin');
21 • insert role (role_name) values ('user');
22
23 • CREATE TABLE file (
24     file_id int NOT NULL AUTO_INCREMENT,
25     cloudinary_file_id varchar(255) NOT NULL,

```
- Action Output:** Shows the execution log with 11 entries, each detailing a query execution time, action, response (number of rows affected), and duration/fetch time.

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



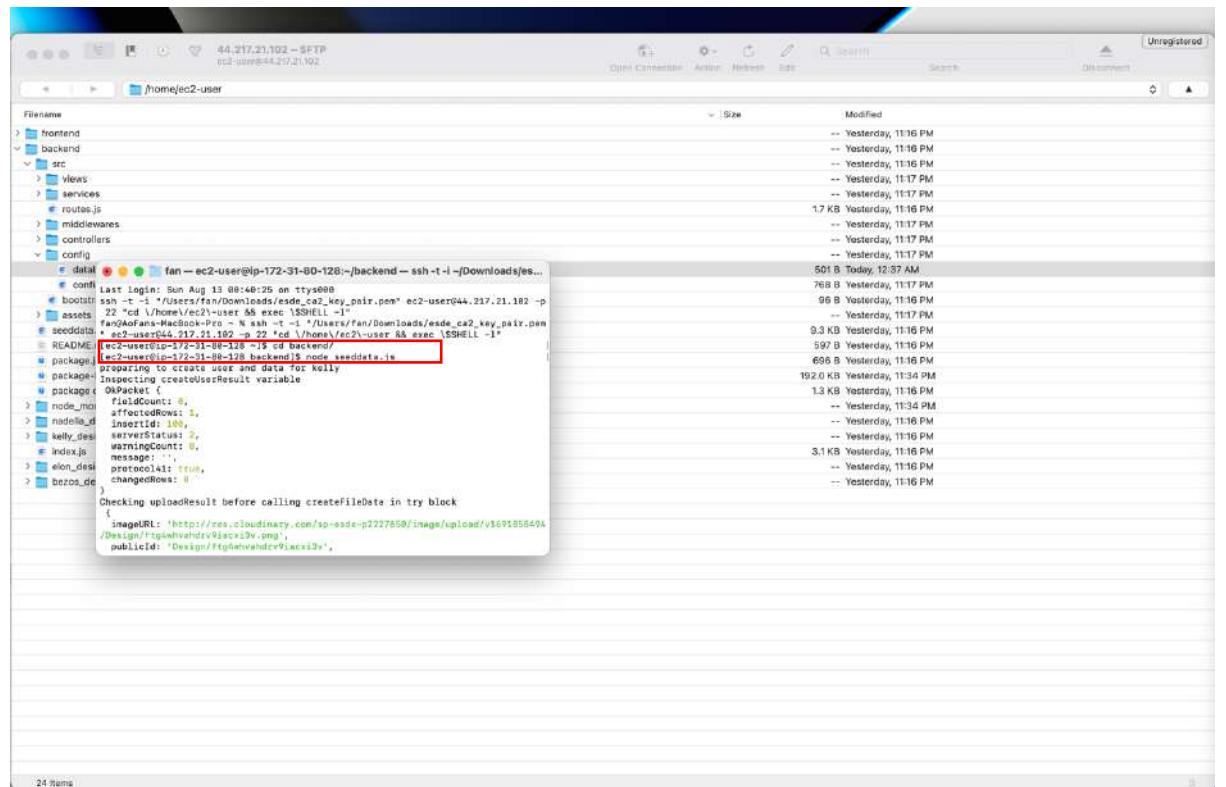
- In Visual Studio Code, replace host with RDS Endpoint.
(line 8)

```

private var folders = 83 > /root/m628928_cy0xq35m000gn > T > chasudo.cybanduck > editor-66dc0d2-a8b5-490c-8058-afb603103a09 > home > ec2-user > backend > src > config > 1876015289 > database.js > ..<
database.js - AY232411ST0505 ESDE CAT Insecure API Code 4<
localhost:~> cat database.js
1 const mysql = require('mysql');
2 const config = require('./config');
3 // To find out more on createPool:
4 // https://www.npmjs.com/package/mysql#pooling-connect
5
6 const pool = mysql.createPool({
7   connectionLimit: 10,
8   host: config.database.host,
9   user: config.database.username,
10  password: config.database.password,
11  database: config.databaseName,
12  multipleStatements: true
13 });
14
15 module.exports = pool;

```

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



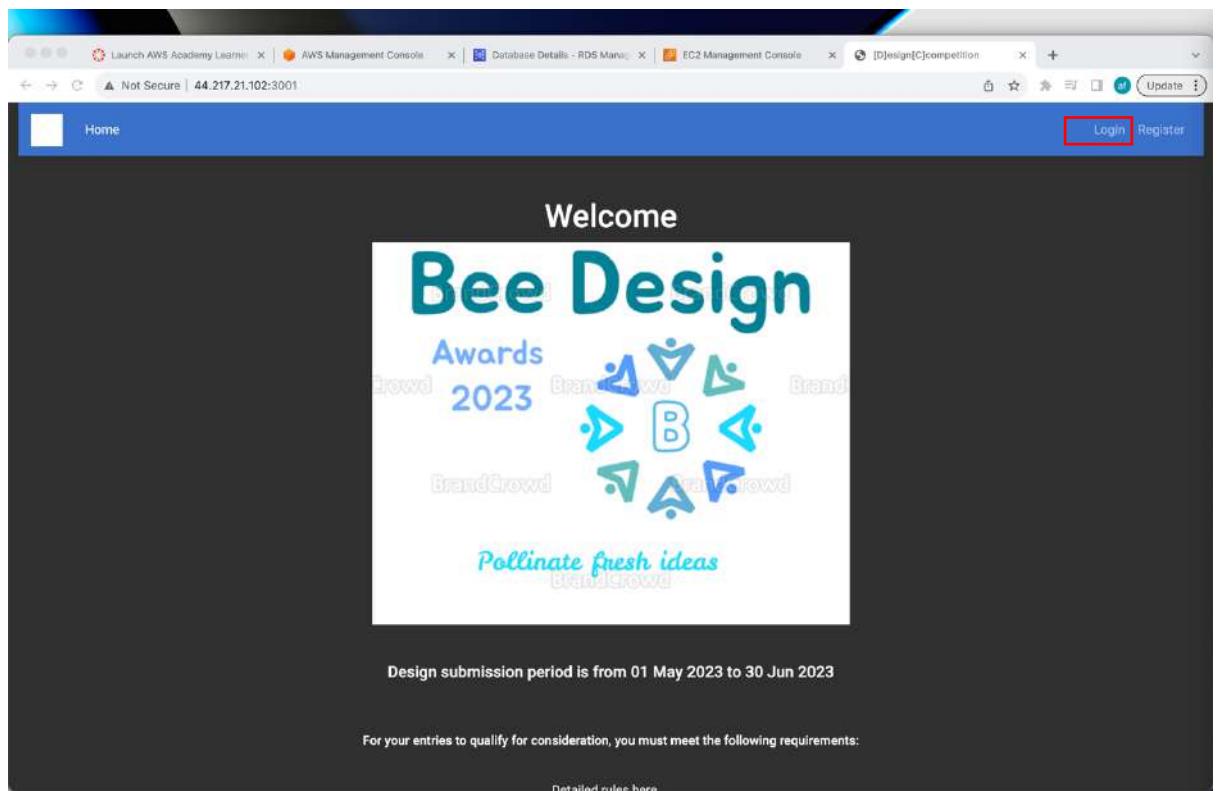
- Open 2 terminals in Cyberduck, cd the front-end and back-end respectively and run npm start.

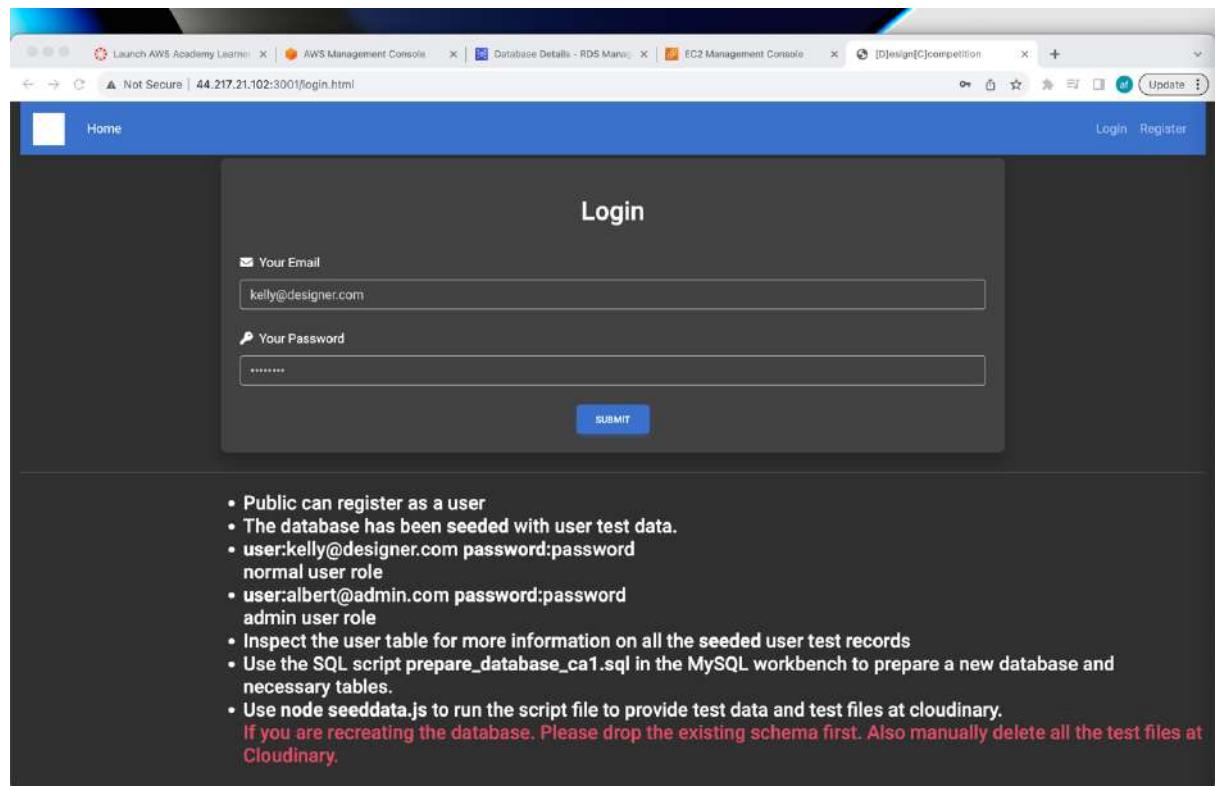
The screenshot shows a terminal window with two panes. The left pane displays the directory structure of the 'frontend' project, which includes subfolders like 'src', 'services', 'middlewares', 'controllers', and 'config'. The right pane shows the directory structure of the 'backend' project, which includes 'src', 'services', 'middlewares', 'controllers', and 'config' subfolders. Both panes show files such as 'index.js', 'bootstrap.js', 'assets', 'seeddata.js', 'README.md', 'package.json', and 'package-lock.json'. In the bottom right corner of the right pane, there is a note: 'Example app listening at http://44.217.21.102:5001'.

```
44.217.21.102 - SFTP
ec2-user@44.217.21.102:
File Connections Action Refresh Edit Search Disconnect
Filename Size Modified
+-- frontend
+-- backend
+-- src
+-- views
+-- services
+-- middlewares
+-- controllers
+-- config
+-- database.js
+-- config.js
+-- bootstrap.js
+-- assets
seeddata.js Last login: Sun Aug 13 00:41:20 on ttys000
README.md ssh -t -i "/Users/fan/Downloads/esdc_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd /Users/fan/Downloads/; exec $SSH_ORIGINAL_COMMAND"
package.json * ec2-user@44.217.21.102 ~ 22 "cd /Users/fan/Downloads/; exec $SSH_ORIGINAL_COMMAND"
package-lock.json [ec2-user@ip-172-31-80-128 ~]$ cd frontend/
[ec2-user@ip-172-31-80-128 frontend]$ npm start
+-- node_modules > firstfrontend@0.0.0 start
+-- radella_desig > nodemon index.js
+-- kelly_design
+-- index.js [nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching directory: .
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server hosted at http://44.217.21.102:3001
[]

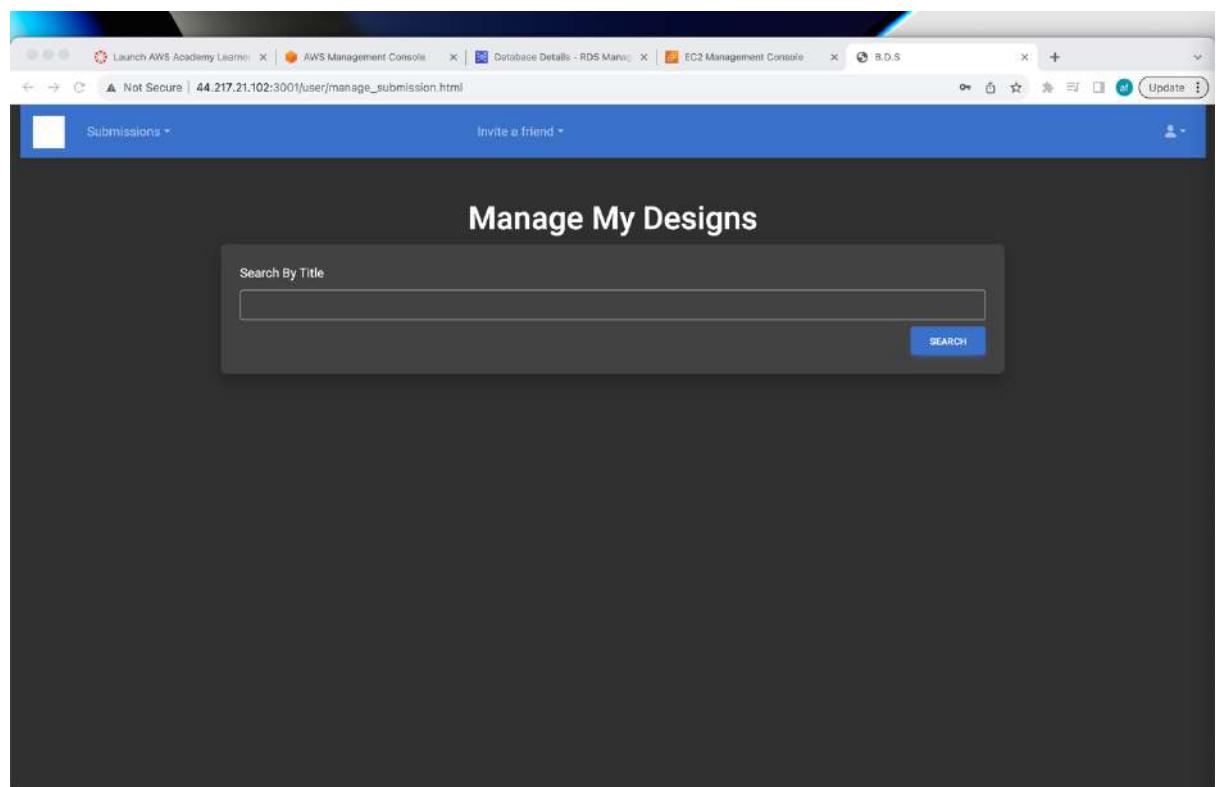
+-- frontend
+-- backend
+-- src
+-- views
+-- services
+-- middlewares
+-- controllers
+-- config
+-- database.js
+-- config.js
+-- bootstrap.js
+-- assets
Last login: Sun Aug 13 00:46:25 on ttys001
ssh -t -i "/Users/fan/Downloads/esdc_ca2_key_pair.pem" ec2-user@44.217.21.102 -p 22 "cd /Users/fan/Downloads/; exec $SSH_ORIGINAL_COMMAND"
fan@ip-172-31-80-128:~$ cd /Users/fan/Downloads/; exec $SSH_ORIGINAL_COMMAND
[ec2-user@ip-172-31-80-128 ~]$ cd backend/
[ec2-user@ip-172-31-80-128 backend]$ npm start
> xyz@1.0.0 start
> nodemon index.js
[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching directory: .
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Example app listening at http://44.217.21.102:5001
[]
```

- 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.



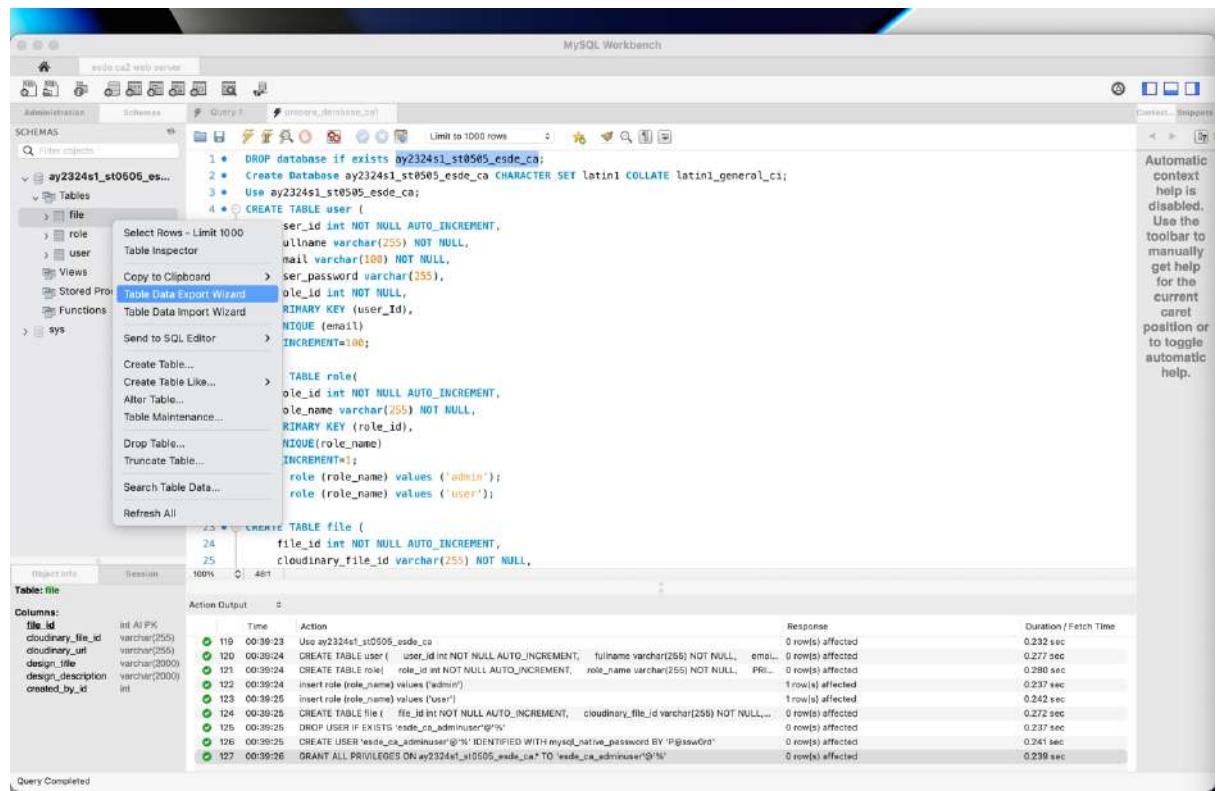


- When successfully logged in, it indicates that the bee design app is connected to the RDS database instance.



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.



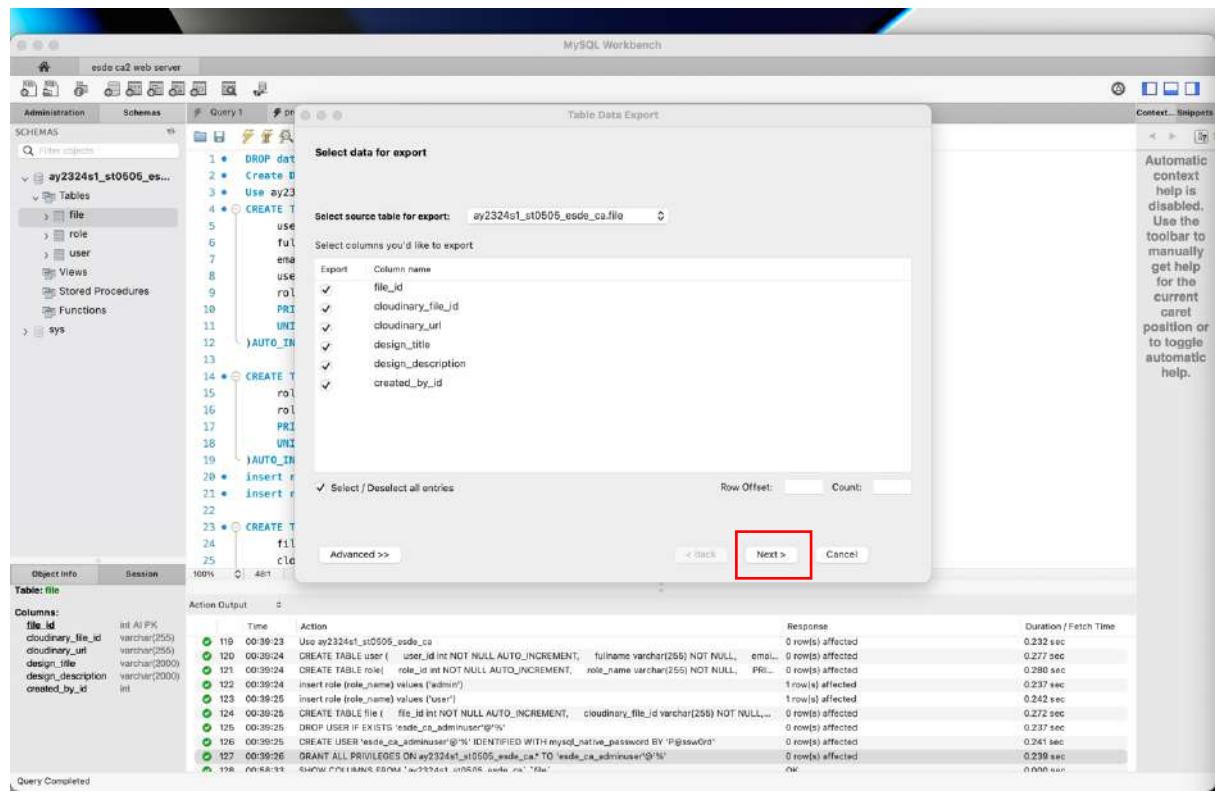
The screenshot shows the MySQL Workbench interface. A context menu is open over a table named 'file'. The menu path 'Table Data Export Wizard' is highlighted. The menu also includes options like 'Select Rows - Limit 1000', 'Table Inspector', 'Copy to Clipboard', 'Send to SQL Editor', and 'Create Table...', among others. The main query editor window displays several SQL statements related to creating databases and tables, including:

```
1 • DROP database if exists ay2324s1_st0505_esde_ca;
2 • Create Database ay2324s1_st0505_esde_ca CHARACTER SET latin1 COLLATE latin1_general_ci;
3 • Use ay2324s1_st0505_esde_ca;
4 • CREATE TABLE user (
    ser_id int NOT NULL AUTO_INCREMENT,
    username varchar(255) NOT NULL,
    mail varchar(100) NOT NULL,
    ser_password varchar(255),
    role_id int NOT NULL,
    PRIMARY KEY (user_Id),
    UNIQUE (email),
    INCREMENT=100;
)
TABLE role(
    role_id int NOT NULL AUTO_INCREMENT,
    role_name varchar(255) NOT NULL,
    PRIMARY KEY (role_id),
    UNIQUE(role_name),
    INCREMENT=1;
)
role (role_name) values ('admin');
role (role_name) values ('user');

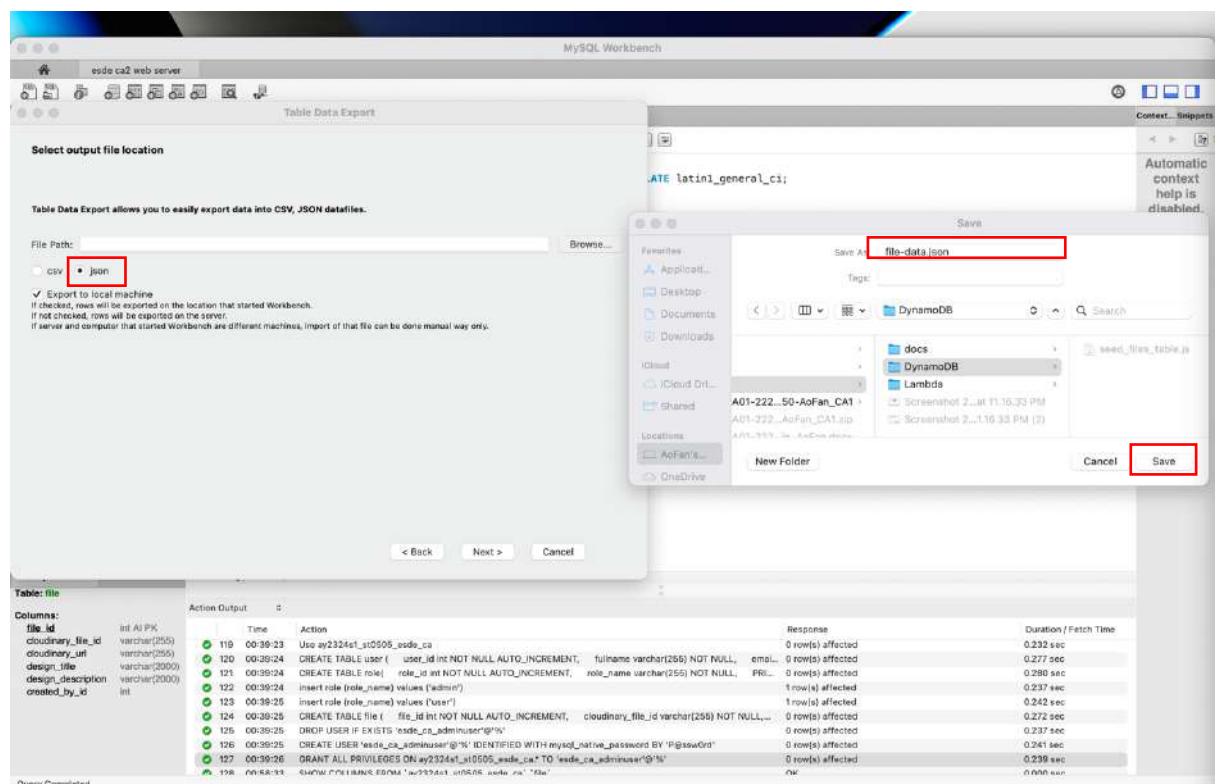
CREATE TABLE file (
    file_id int NOT NULL AUTO_INCREMENT,
    clouddinary_file_id varchar(255) NOT NULL,
```

The 'Action Output' tab at the bottom shows the execution log with 127 entries, indicating successful creation of the database, tables, and users.

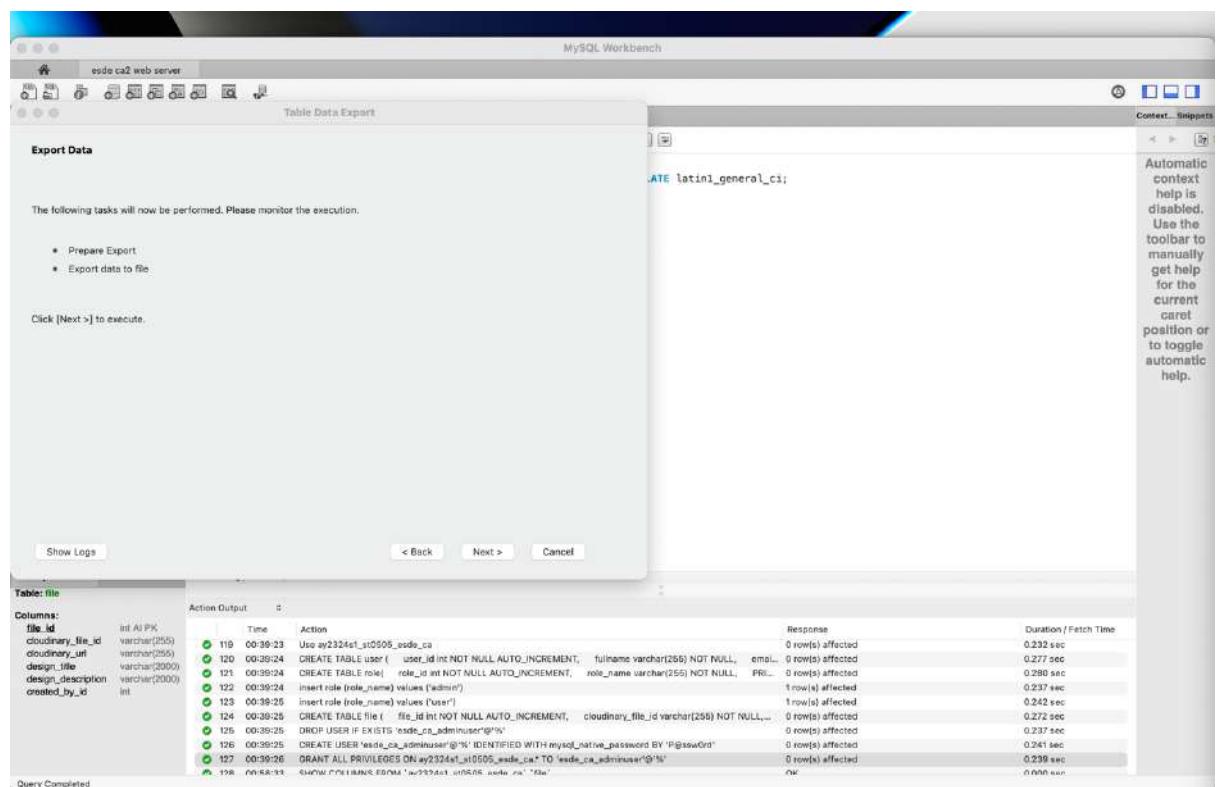
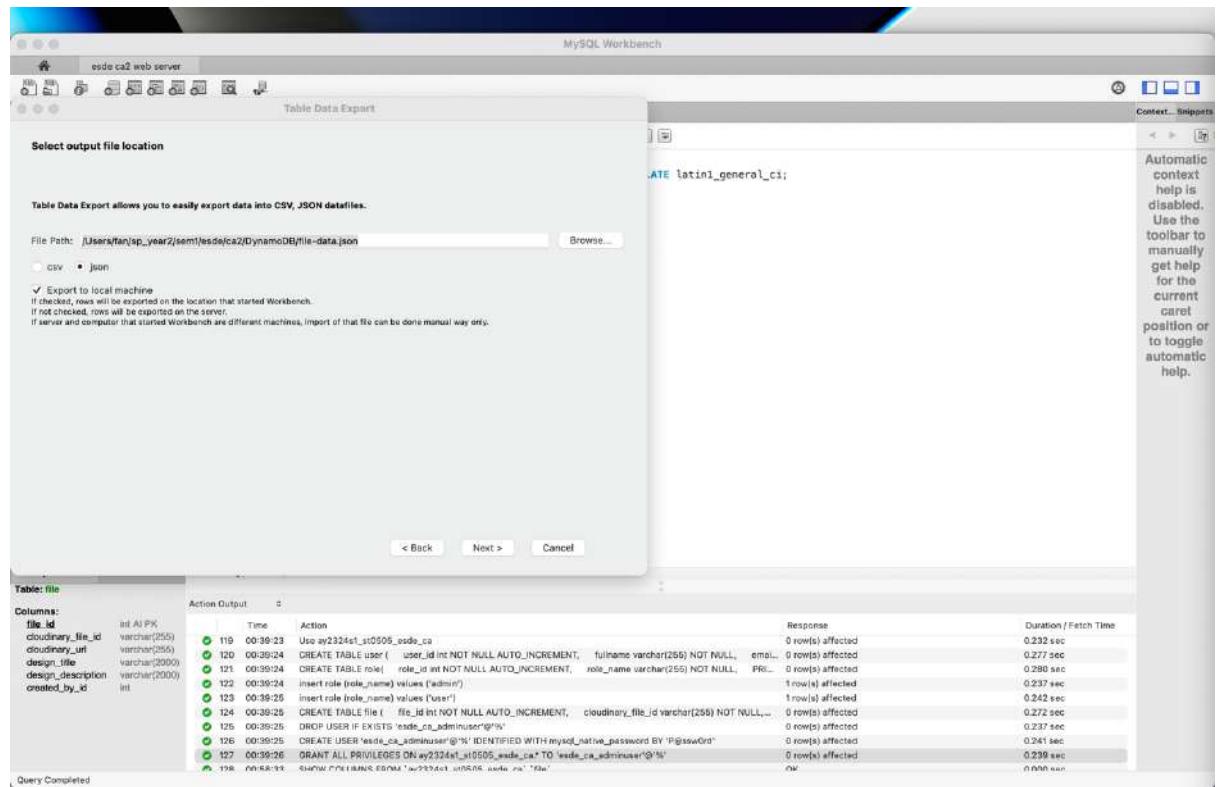
- On Table Data Export tab, click Next >.



- 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.



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

Action Output

Time	Action	Response	Duration / Fetch Time
119 00:39:23	Use ay2324s1_st0505_esde_ca	0 row(s) affected	0.232 sec
120 00:39:24	CREATE TABLE user (`user_id` int NOT NULL AUTO_INCREMENT, `fullname` varchar(256) NOT NULL, `email` varchar(256) NOT NULL, `role_id` int NOT NULL AUTO_INCREMENT, `role_name` varchar(256) NOT NULL, `privileges` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.277 sec
121 00:39:24	CREATE TABLE role (`role_id` int NOT NULL AUTO_INCREMENT, `role_name` varchar(256) NOT NULL, `privileges` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.280 sec
122 00:39:24	insert role (role_name values ('admin'))	1 row(s) affected	0.237 sec
123 00:39:25	insert role (role_name values ('user'))	1 row(s) affected	0.242 sec
124 00:39:25	CREATE TABLE file (`file_id` int NOT NULL AUTO_INCREMENT, `cloudinary_file_id` varchar(256) NOT NULL, `design_file` varchar(2000) NOT NULL, `design_description` varchar(2000) NOT NULL, `created_by_id` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.272 sec
125 00:39:25	DROP USER IF EXISTS 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.237 sec
126 00:39:25	CREATE USER 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.241 sec
127 00:39:26	GRANT ALL PRIVILEGES ON ay2324s1_st0505_esde_ca.* TO 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.239 sec
128 nn:48:11	flush privileges	0 row(s) affected	0.000 sec

Query Completed

MySQL Workbench

Table Data Export

Export Results

File C:/Users/fan/Downloads/file-data.json was exported in 0.628 s

Exported 36 records

Action Output

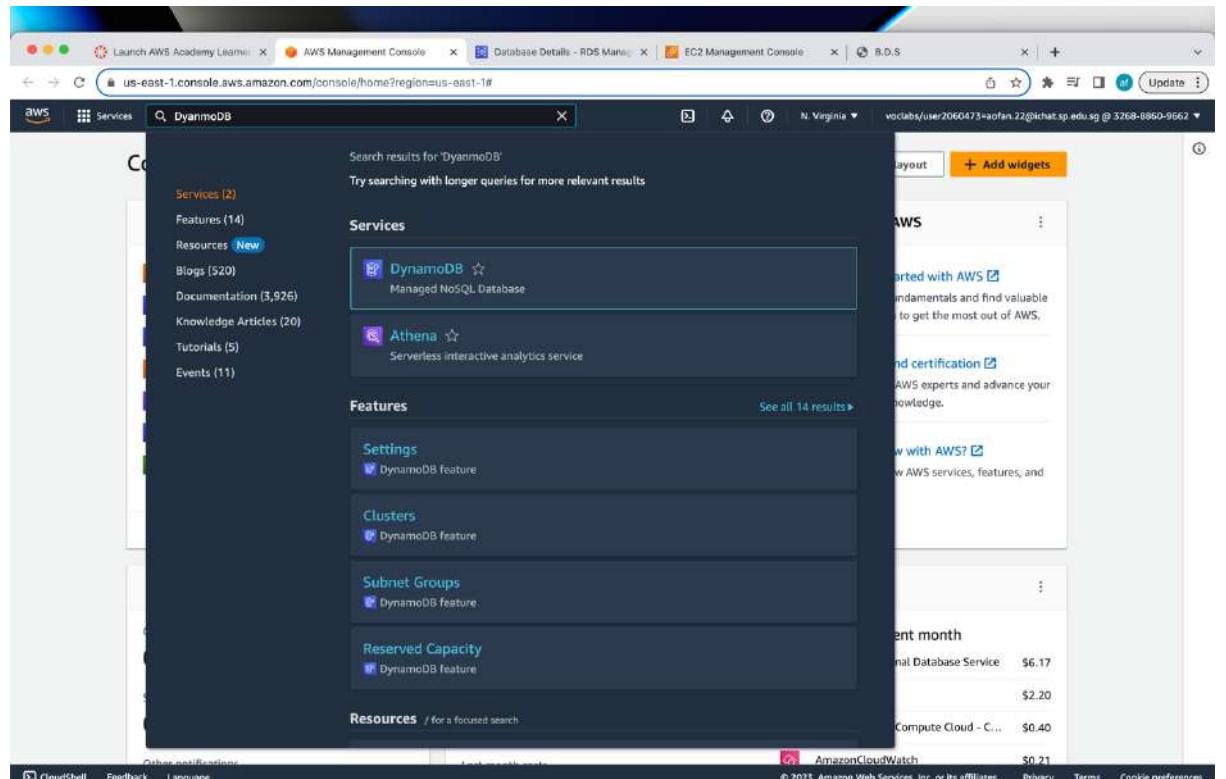
Time	Action	Response	Duration / Fetch Time
119 00:39:23	Use ay2324s1_st0505_esde_ca	0 row(s) affected	0.232 sec
120 00:39:24	CREATE TABLE user (`user_id` int NOT NULL AUTO_INCREMENT, `fullname` varchar(256) NOT NULL, `email` varchar(256) NOT NULL, `role_id` int NOT NULL AUTO_INCREMENT, `role_name` varchar(256) NOT NULL, `privileges` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.277 sec
121 00:39:24	CREATE TABLE role (`role_id` int NOT NULL AUTO_INCREMENT, `role_name` varchar(256) NOT NULL, `privileges` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.280 sec
122 00:39:24	insert role (role_name values ('admin'))	1 row(s) affected	0.237 sec
123 00:39:25	insert role (role_name values ('user'))	1 row(s) affected	0.242 sec
124 00:39:25	CREATE TABLE file (`file_id` int NOT NULL AUTO_INCREMENT, `cloudinary_file_id` varchar(256) NOT NULL, `design_file` varchar(2000) NOT NULL, `design_description` varchar(2000) NOT NULL, `created_by_id` int NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1_general_ci;	0 row(s) affected	0.272 sec
125 00:39:25	DROP USER IF EXISTS 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.237 sec
126 00:39:25	CREATE USER 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.241 sec
127 00:39:26	GRANT ALL PRIVILEGES ON ay2324s1_st0505_esde_ca.* TO 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@sswOrD'	0 row(s) affected	0.239 sec
128 nn:48:11	flush privileges	0 row(s) affected	0.000 sec

Query Completed

- The file exported should look like this in TextEdit.

```
[{"file_id":100, "cloudinary_file_id": "Design/ftg4vhahdrv9iacxLb", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ftg4vhahdrv9iacx19.png", "design_title": "kelly design 1", "design_description": "kelly design 1 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":101, "cloudinary_file_id": "Design/btuvfhrjjeyp4ewm", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/btuvfhrjjeyp4ewm.png", "design_title": "kelly design 2", "design_description": "kelly design 2 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":102, "cloudinary_file_id": "Design/ekqzqy0w08lq1g", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ekqzqy0w08lq1g.png", "design_title": "kelly design 3", "design_description": "kelly Design 3 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":103, "cloudinary_file_id": "Design/elat9twtwqjyekz92", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/elat9twtwqjyekz92.png", "design_title": "kelly design 4", "design_description": "kelly design 4 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":104, "cloudinary_file_id": "Design/udpn19ksxytvdn", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/udpn19ksxytvdn.png", "design_title": "kelly design 5", "design_description": "kelly Design 5 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":105, "cloudinary_file_id": "Design/cvewoye0sko7nxe4", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/cvewoye0sko7nxe4.png", "design_title": "kelly design 6", "design_description": "kelly design 6 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":106, "cloudinary_file_id": "Design/b03khze5ejghpoyavk", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/b03khze5ejghpoyavk.png", "design_title": "kelly design 7", "design_description": "kelly Design 7 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":107, "cloudinary_file_id": "Design/cosScsuu0kl5vcigpr", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/cosScsuu0kl5vcigpr.png", "design_title": "kelly design 8", "design_description": "kelly design 8 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":108, "cloudinary_file_id": "Design/lazcmxzfmwp2ptysig", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/lazcmxzfmwp2ptysig.png", "design_title": "kelly design 9", "design_description": "kelly design 9 description text 1 text 2 text 3 text 4 ....", "created_by": "id_100"}, {"file_id":109, "cloudinary_file_id": "Design/nfwphsnnc5tixhgc", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/nfwphsnnc5tixhgc.png", "design_title": "bezos design 1", "design_description": "bezos design 1 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":110, "cloudinary_file_id": "Design/mktcnmpnuc105c5e", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/mktcnmpnuc105c5e.png", "design_title": "bezos design 2", "design_description": "bezos design 2 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":111, "cloudinary_file_id": "Design/aximahjpebiksd1ge", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/aximahjpebiksd1ge.png", "design_title": "bezos design 3", "design_description": "bezos Design 3 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":112, "cloudinary_file_id": "Design/w3ky3uqzb0kmicud", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/w3ky3uqzb0kmicud.png", "design_title": "bezos design 4", "design_description": "bezos design 4 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":113, "cloudinary_file_id": "Design/e3ggwfyjyhdB29nmnpn", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/e3ggwfyjyhdB29nmnpn.png", "design_title": "bezos design 5", "design_description": "bezos Design 5 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":114, "cloudinary_file_id": "Design/luiu0nxmjktxf269cb", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/luiu0nxmjktxf269cb.png", "design_title": "bezos design 6", "design_description": "bezos design 6 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":115, "cloudinary_file_id": "Design/uytf7cl1urstue02", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/uytf7cl1urstue02.png", "design_title": "bezos design 7", "design_description": "bezos Design 7 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":116, "cloudinary_file_id": "Design/y2femzefbxsw0f", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/y2femzefbxsw0f.png", "design_title": "bezos design 8", "design_description": "bezos design 8 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":117, "cloudinary_file_id": "Design/zvag6ftjopahrwsf12", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/zvag6ftjopahrwsf12.png", "design_title": "bezos design 9", "design_description": "bezos Design 9 description text 1 text 2 text 3 text 4 ....", "created_by": "id_110"}, {"file_id":118, "cloudinary_file_id": "Design/ih6fju3uU7uzinghwc", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ih6fju3uU7uzinghwc.png", "design_title": "elon design 1", "design_description": "elon design 1 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":119, "cloudinary_file_id": "Design/beehrsvoilaMB88Bv1c72", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/beehrsvoilaMB88Bv1c72.png", "design_title": "elon design 2", "design_description": "elon design 2 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":120, "cloudinary_file_id": "Design/bckbdw7tveqpd5d", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/bckbdw7tveqpd5d.png", "design_title": "elon design 3", "design_description": "elon design 3 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":121, "cloudinary_file_id": "Design/uytdywh33wv5440r", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/uytdywh33wv5440r.png", "design_title": "elon design 4", "design_description": "elon design 4 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":122, "cloudinary_file_id": "Design/ds1u0wkw7kmnxf5h0q", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ds1u0wkw7kmnxf5h0q.png", "design_title": "elon design 5", "design_description": "elon design 5 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":123, "cloudinary_file_id": "Design/lok255ghedw5hfb0d", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/lok255ghedw5hfb0d.png", "design_title": "elon design 6", "design_description": "elon design 6 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":124, "cloudinary_file_id": "Design/dtk2pcipiriv2vhnqf2", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/dtk2pcipiriv2vhnqf2.png", "design_title": "elon design 7", "design_description": "elon design 7 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":125, "cloudinary_file_id": "Design/15n6gpmzqnbwqudet", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/15n6gpmzqnbwqudet.png", "design_title": "elon design 8", "design_description": "elon design 8 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":126, "cloudinary_file_id": "Design/uytf7cl1urstue02", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/uytf7cl1urstue02.png", "design_title": "elon design 9", "design_description": "elon Design 9 description text 1 text 2 text 3 text 4 ....", "created_by": "id_102"}, {"file_id":127, "cloudinary_file_id": "Design/w8hjhphohngpt", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/w8hjhphohngpt.png", "design_title": "nadeela design 1", "design_description": "nadeela design 1 description text 1 text 2 text 3 text 4 ....", "created_by": "id_103"}, {"file_id":128, "cloudinary_file_id": "Design/ufhjqdyqjwpoem4pk", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ufhjqdyqjwpoem4pk.png", "design_title": "nadeela design 2", "design_description": "nadeela design 2 description text 1 text 2 text 3 text 4 ....", "created_by": "id_103"}, {"file_id":129, "cloudinary_file_id": "Design/ufhjqdyqjwpoem4pk", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/ufhjqdyqjwpoem4pk.png", "design_title": "nadeela design 3", "design_description": "nadeela design 3 description text 1 text 2 text 3 text 4 ....", "created_by": "id_103"}, {"file_id":130, "cloudinary_file_id": "Design/uf44cmk3qg8ml0h", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/uf44cmk3qg8ml0h.png", "design_title": "nadeela design 4", "design_description": "nadeela design 4 description text 1 text 2 text 3 text 4 ....", "created_by": "id_103"}, {"file_id":131, "cloudinary_file_id": "Design/u44cmk3qg8ml0h", "cloudinary_url": "http://res.cloudinary.com/csp-esde-02227650/image/upload/v1691858494/Design/u44cmk3qg8ml0h.png", "design_title": "nadeela design 5", "design_description": "nadeela design 5 description text 1 text 2 text 3 text 4 ....", "created_by": "id_103"}]
```

- Navigate to DynamoDB via AWS Management Console.



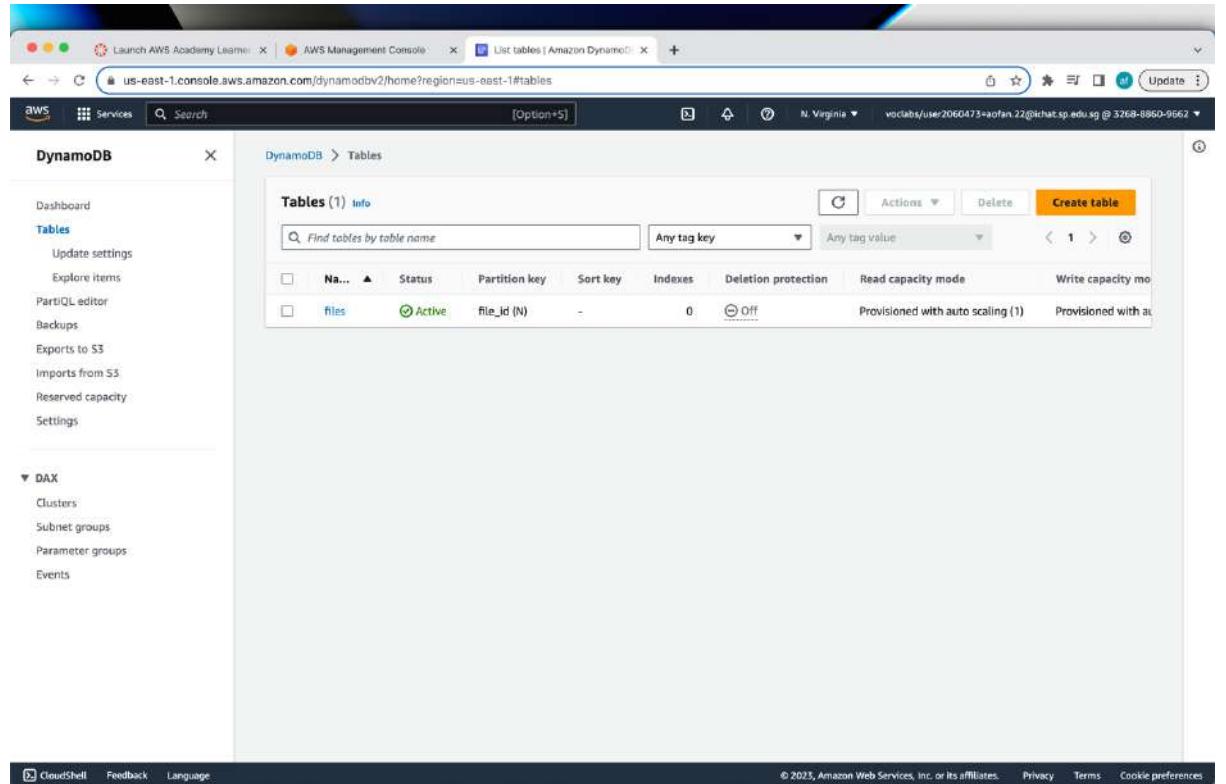
- Click Create table.

The screenshot shows the AWS Management Console with the URL us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#dashboard. The left sidebar is open, showing 'DynamoDB' selected under 'Tables'. The main area is titled 'Dashboard' and contains sections for 'Alarms (0)', 'DAX clusters (0)', and 'Create resources'. The 'Create resources' section has a large orange button labeled 'Create table' which is highlighted with a red box. To the right of the button, there is a brief description of Amazon DynamoDB Accelerator (DAX) and a link to learn more.

- 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.

The screenshot shows the 'Create table' wizard in the AWS Management Console. The top navigation bar includes tabs for 'Launch AWS Academy', 'AWS Management Console', 'Dashboard | Amazon DynamoDB', 'Database Details - RDS MySQL', 'EC2 Management Console', 'B.D.S.', and 'CloudShell'. The main content area is titled 'Create table' and has two main sections: 'Table details' and 'Table settings'. In the 'Table details' section, the 'Table name' is set to 'files' and the 'Partition key' is set to 'file_id' with a dropdown menu showing 'Number'. In the 'Table settings' section, there are two options: 'Default settings' (selected) and 'Customize settings'. Both options have small descriptions below them. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and 'Language'.

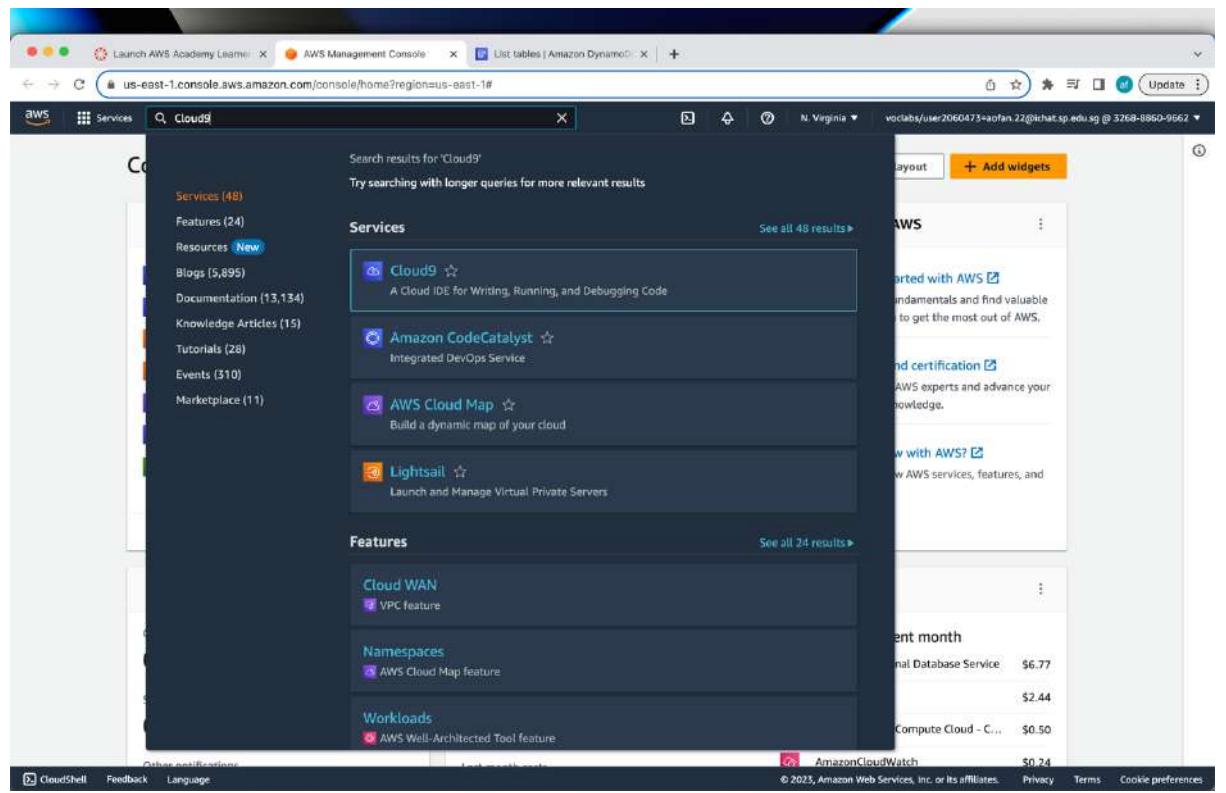
- Head to DynamoDB Tables tab to check the newly created table.



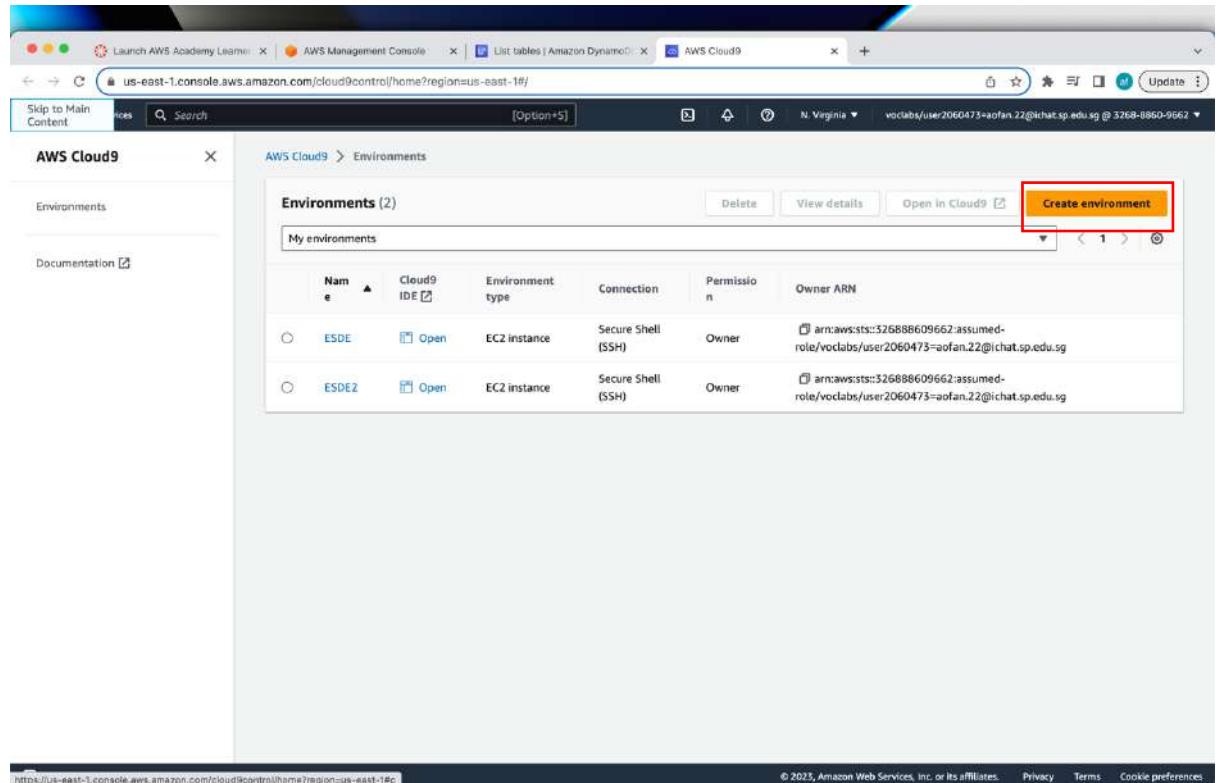
The screenshot shows the AWS Management Console interface for Amazon DynamoDB. The top navigation bar includes tabs for Launch AWS Academy, AWS Management Console, and List tables | Amazon DynamoDB. The URL in the address bar is us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#tables. The sidebar on the left is titled "DynamoDB" and contains links for Dashboard, Tables (which is selected), Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, and Settings. Below this is a section for "DAX" with links for Clusters, Subnet groups, Parameter groups, and Events. The main content area is titled "Tables (1) info" and displays a table with one row. The table columns are: Name, Status, Partition key, Sort key, Indexes, Deletion protection, Read capacity mode, and Write capacity mode. The single table entry is named "files", has a status of "Active", a partition key of "file_id (N)", and deletion protection set to "Off". It also indicates "Provisioned with auto scaling (1)" for both read and write capacity. The bottom of the page includes standard AWS footer links for CloudShell, Feedback, Language, and legal notices like © 2023, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, and Cookie preferences.

Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode
files	Active	file_id (N)	-	0	Off	Provisioned with auto scaling (1)	Provisioned with auto scaling (1)

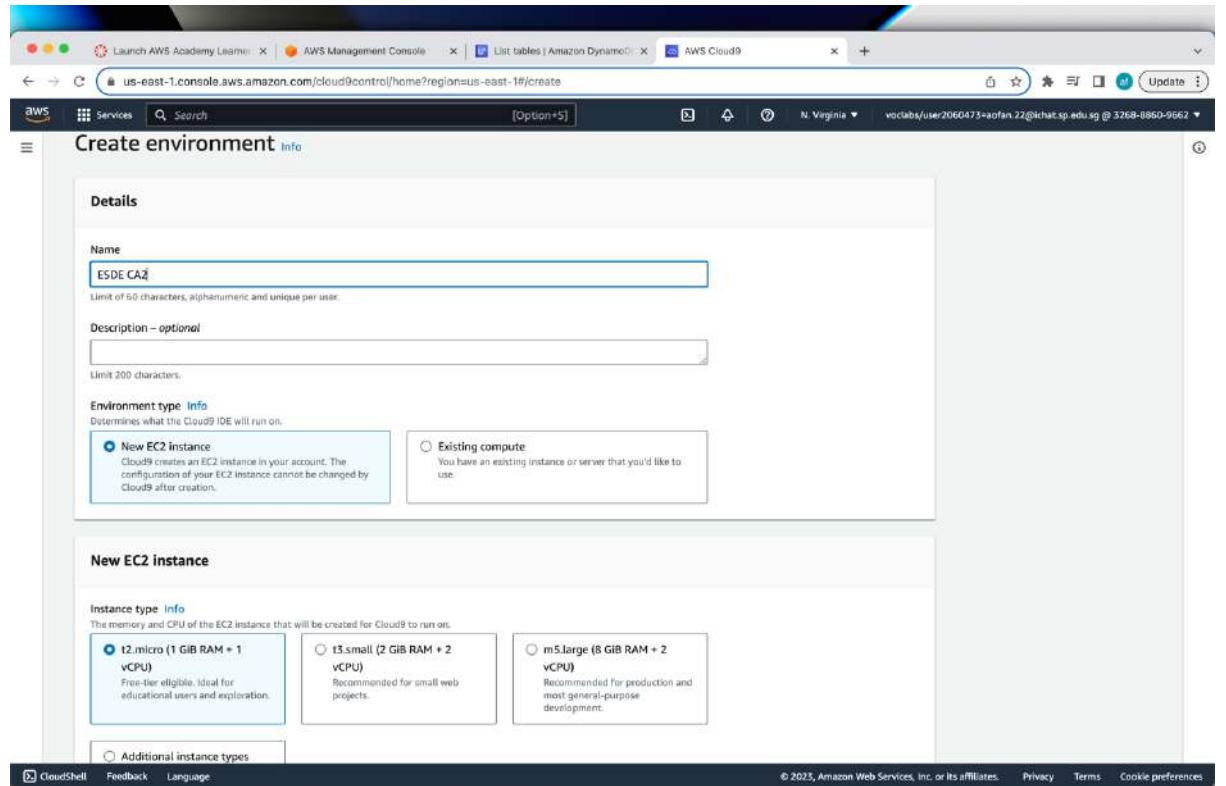
- Navigate to Cloud9



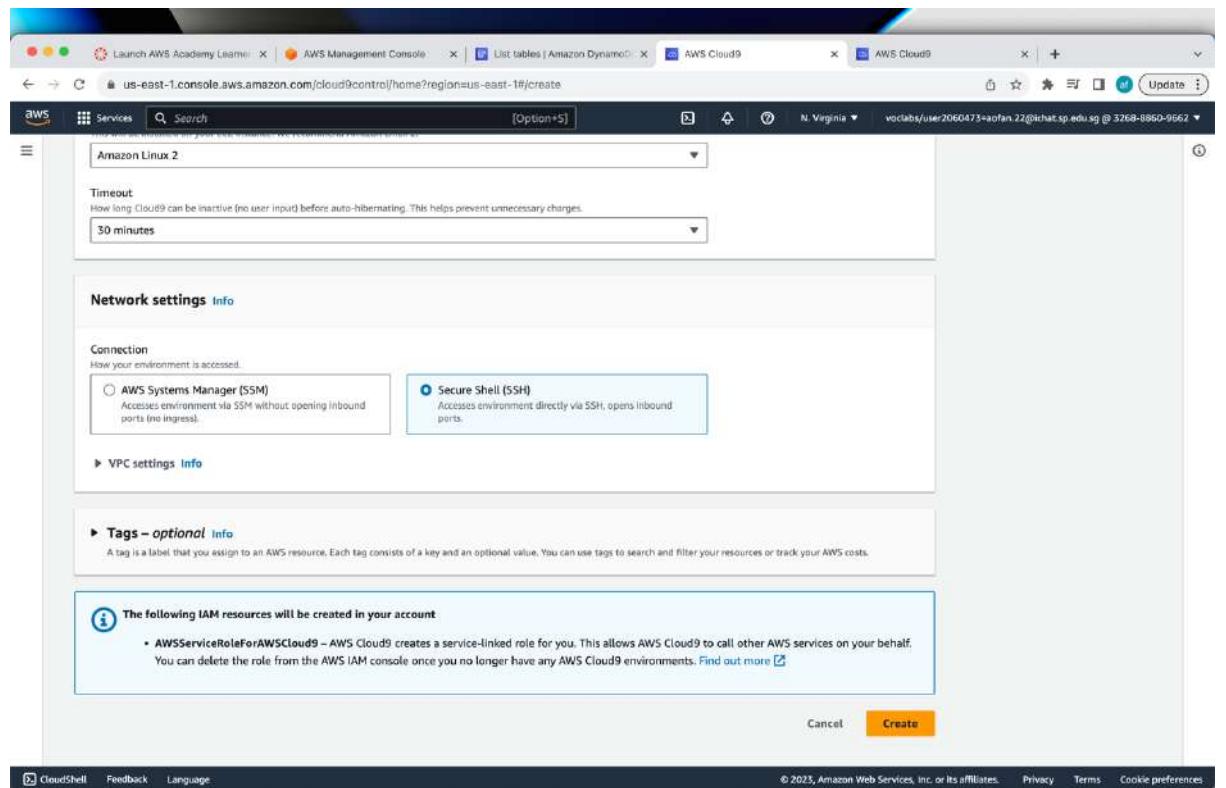
- Click on the Create environment button



- 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.

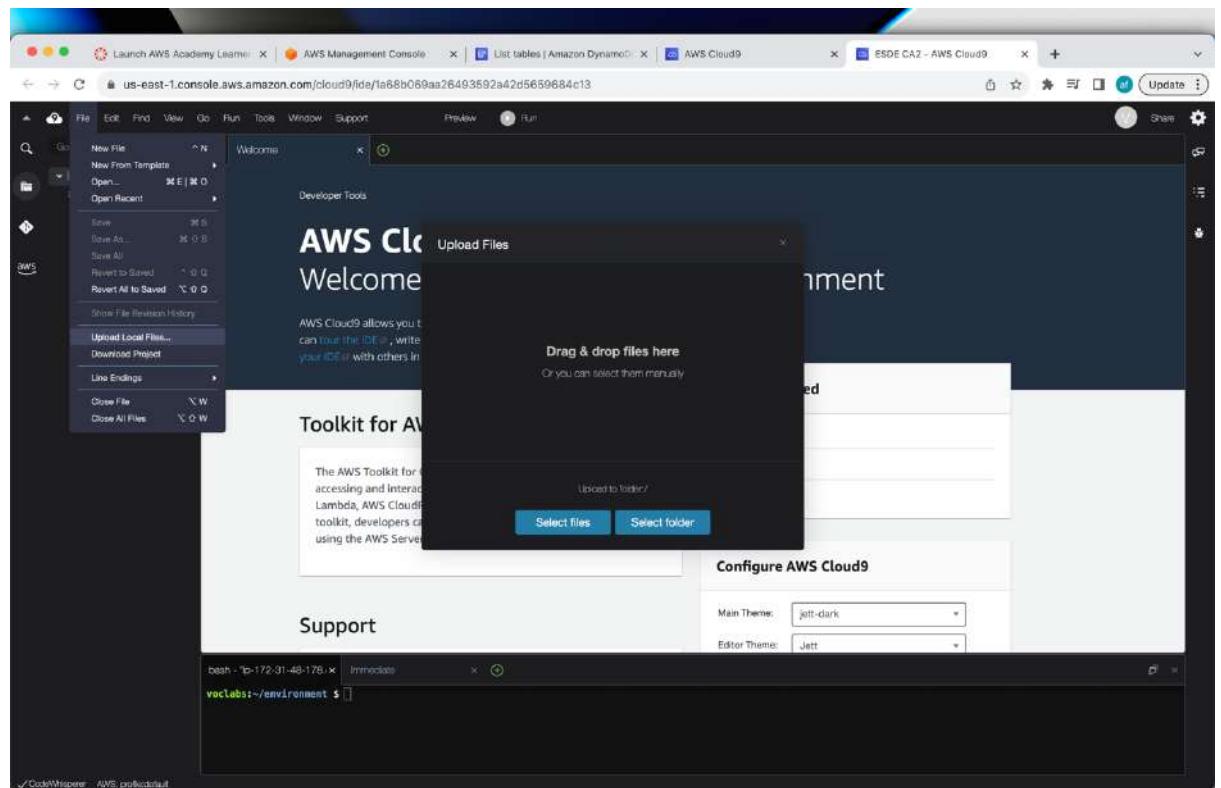


- You will be redirected back the AWS Cloud9 Environment, wait for the environment to set up then click Open.

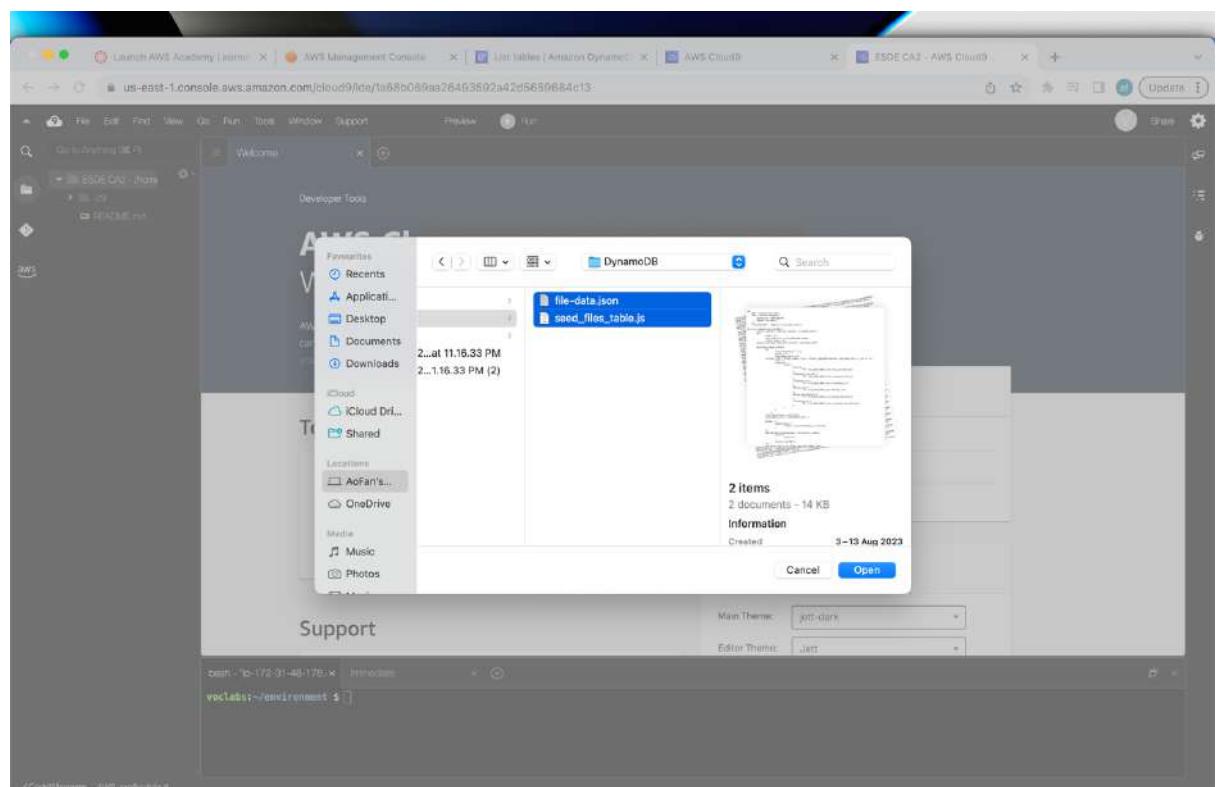
The screenshot shows the AWS Cloud9 console interface. The left sidebar has 'Environments' selected. The main content area displays a table titled 'Environments (3)'. The columns are 'Name', 'Cloud9 IDE', 'Environment type', 'Connection', 'Permission', and 'Owner ARN'. The rows show three environments: 'ESDE' (Cloud9 IDE icon is 'Open'), 'ESDE CA2' (Cloud9 IDE icon is 'Open'), and 'ESDE2' (Cloud9 IDE icon is 'Open'). All environments are of type 'EC2 instance', connected via 'Secure Shell (SSH)', and owned by 'Owner' with the ARN 'arn:aws:sts::326888609662:assumed-role/voclabs/user2060473=aofan.22@ichat.sp.edu.sg'. A progress bar at the top indicates 'Creating ESDE CA2. This can take several minutes. While you wait, see Best practices for using AWS Cloud9.'

This screenshot is identical to the one above, showing the AWS Cloud9 environments list. However, a red rectangular box highlights the 'Open' button for the second environment, 'ESDE CA2'. The rest of the interface and data are the same.

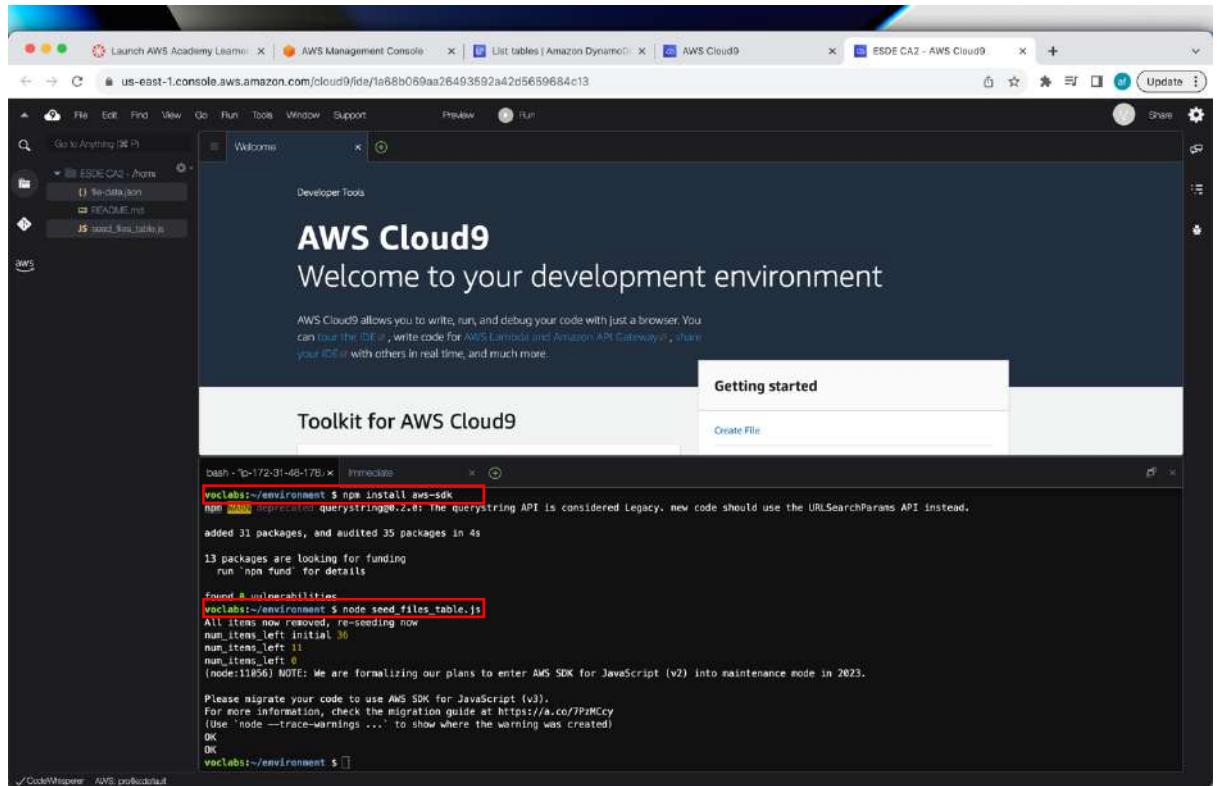
- Select File then click on Upload Local Files.... Upload Files prompt will appear, click on Select files.



- Select file-data.json and seed_files_table.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.



The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file tree with files like `file-data.json`, `README.md`, and `seed_files_table.js`. The main area displays the AWS Cloud9 welcome screen with the message "Welcome to your development environment". Below it is the "Toolkit for AWS Cloud9" section. A terminal window is open with the following command and output:

```

bash - <id-172-31-46-17> ✘ Immediate
vclabs:~/environment $ npm install aws-sdk
npm WARN deprecated querystring@2.0.1: The querystring API is considered Legacy, new code should use the URLSearchParams API instead.

added 31 packages, and audited 35 packages in 4s

13 packages are looking for funding
  run `npm fund` for details

Found 4 vulnerabilities
vclabs:~/environment $ node seed_files_table.js
All items now received, re-seeding now
num_items_left: initial_36
num_items_left: 11
num_items_left: 0
(node:11856) [DEP0005] DeprecationWarning: Requiring non-relative module 'aws-sdk' via require() is deprecated. Use relative module paths instead.
Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the migration guide at https://a.co/7PzKCCy
(Use 'node --trace-warnings ...' to show where the warning was created)
OK
OK
vclabs:~/environment $ 

```

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

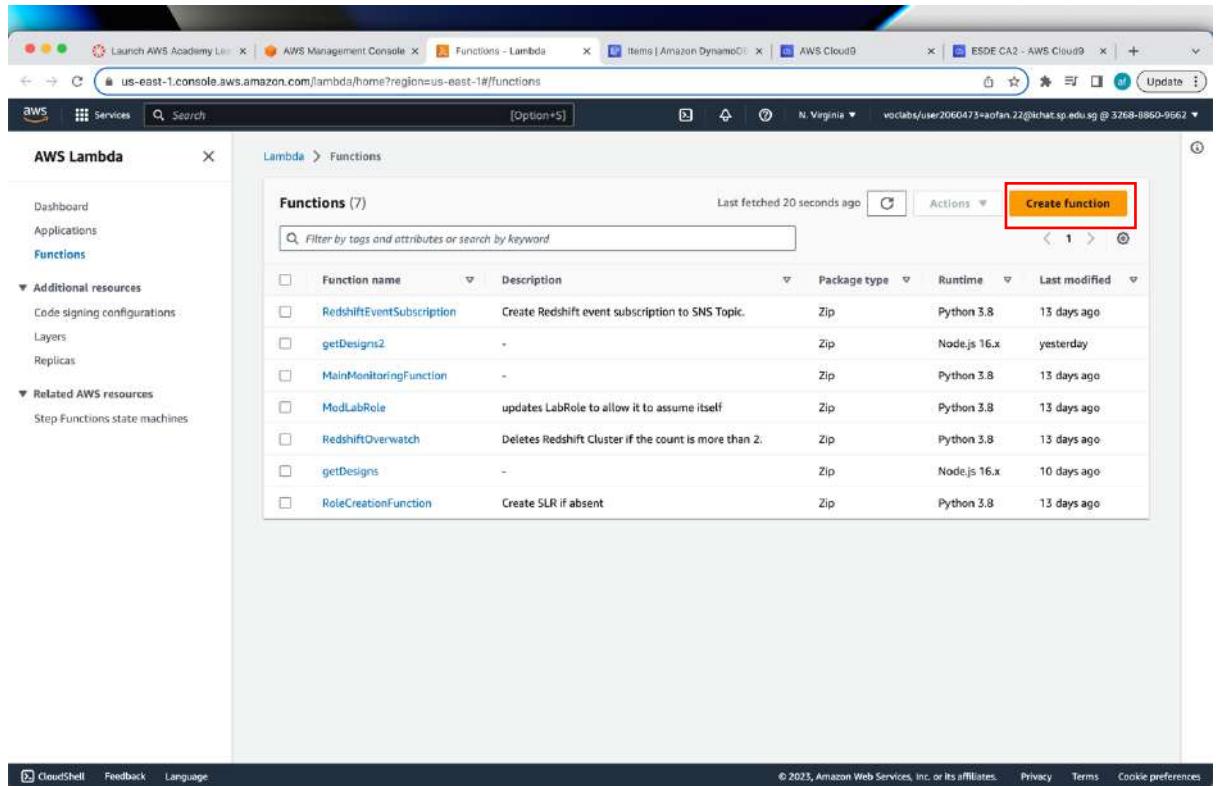
The screenshot shows the AWS DynamoDB console interface. On the left, the navigation pane for 'DynamoDB' is visible, with 'Explore items' highlighted. The main area displays the 'files' table, which contains one item. The 'Scan or query items' section shows the 'Scan' button selected. The 'Items returned' section displays the following data:

file_id (Number)	cloudinary_file_id	cloudinary_url	created_by_id	de
115	Design/nptscj3ltzirstu...	http://res.cloudin...	101	be

A red box highlights the 'Explore items' link in the navigation pane, the 'files' table entry in the list, and the entire 'Items returned' table.

7. Creating and testing ‘getDesginsFunction’ Lambda function

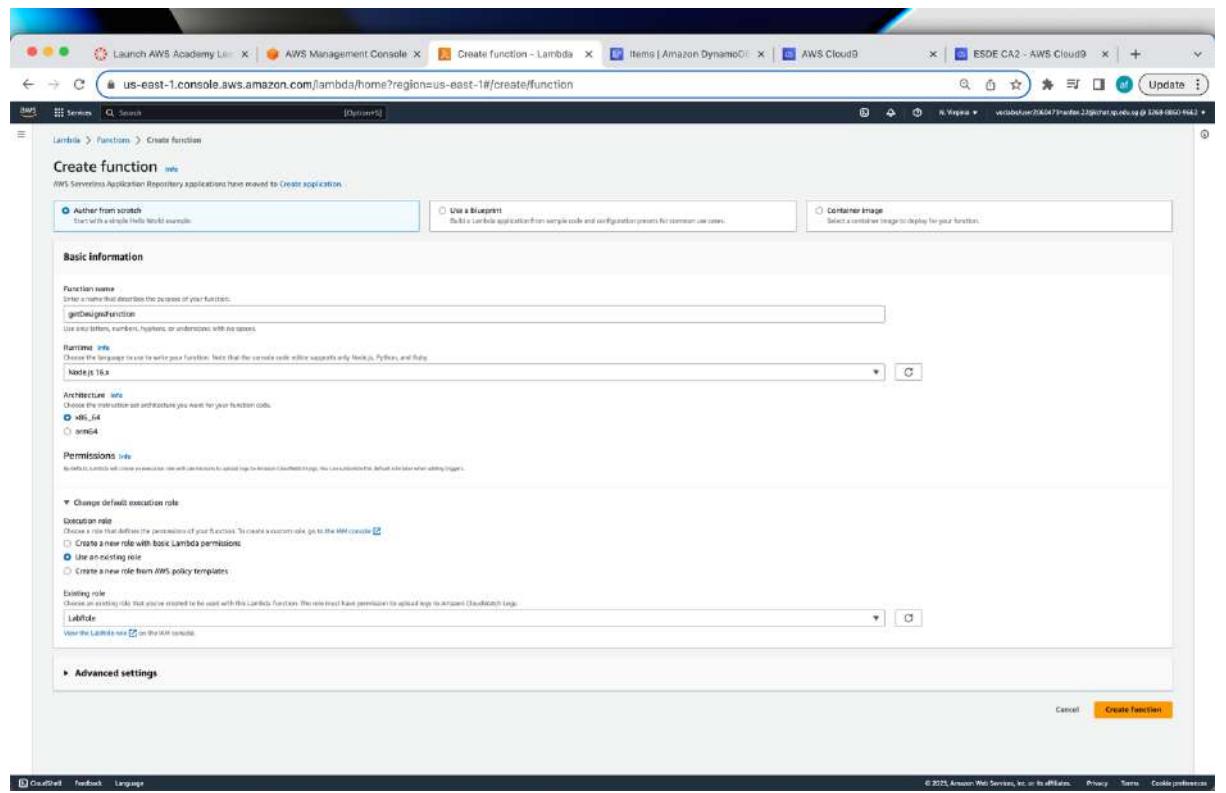
- Navigate to Lambda service console and click Create function.



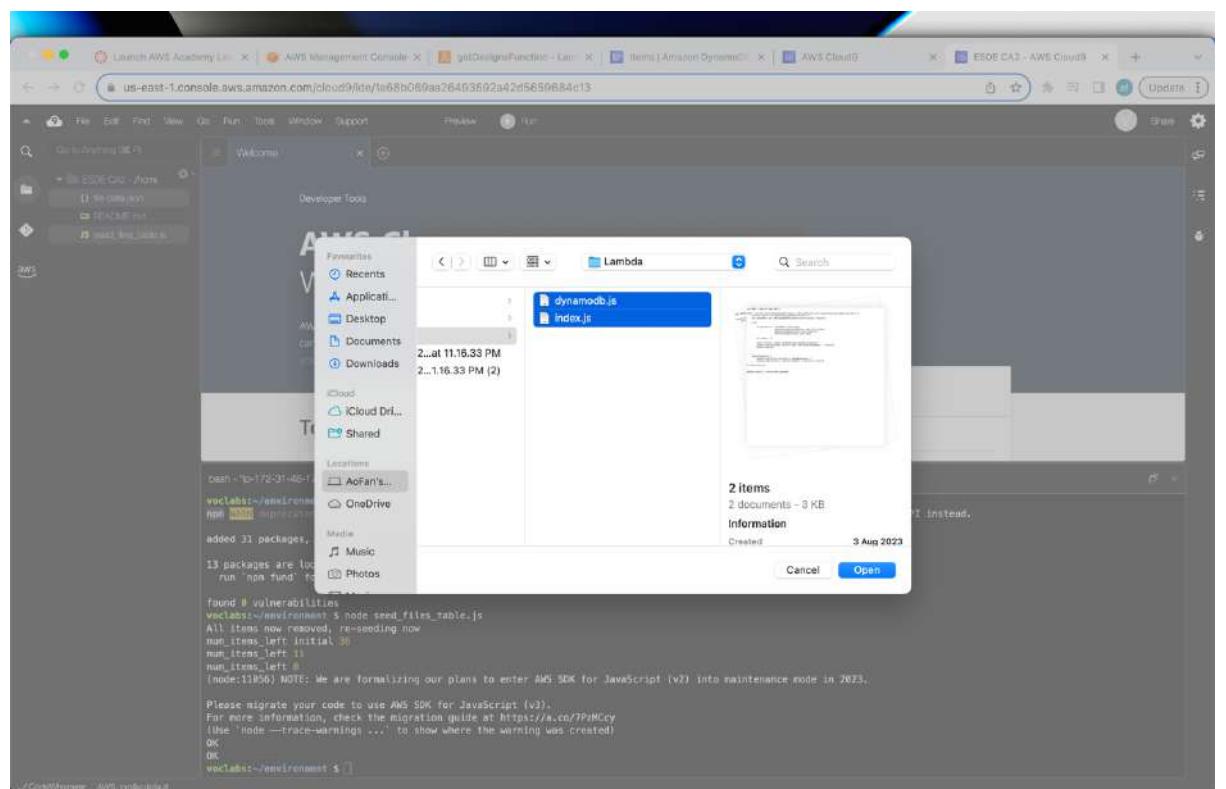
The screenshot shows the AWS Lambda service console. On the left, there's a sidebar with options like Dashboard, Applications, Functions (which is selected and highlighted in blue), Additional resources, and Related AWS resources. The main area is titled 'Functions (7)' and displays a table of existing Lambda functions. The columns in the table are Function name, Description, Package type, Runtime, and Last modified. The table lists seven functions: RedshiftEventSubscription, getDesigns2, MainMonitoringFunction, ModLabRole, RedshiftOverwatch, getDesigns, and RoleCreationFunction. All functions are listed as Zip packages in Python 3.8 runtime. A large orange button labeled 'Create function' is located at the top right of the table area, which is highlighted with a red box.

Function name	Description	Package type	Runtime	Last modified
RedshiftEventSubscription	Create Redshift event subscription to SNS Topic.	Zip	Python 3.8	13 days ago
getDesigns2	-	Zip	Node.js 16.x	yesterday
MainMonitoringFunction	-	Zip	Python 3.8	13 days ago
ModLabRole	updates LabRole to allow it to assume itself	Zip	Python 3.8	13 days ago
RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	13 days ago
getDesigns	-	Zip	Node.js 16.x	10 days ago
RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	13 days ago

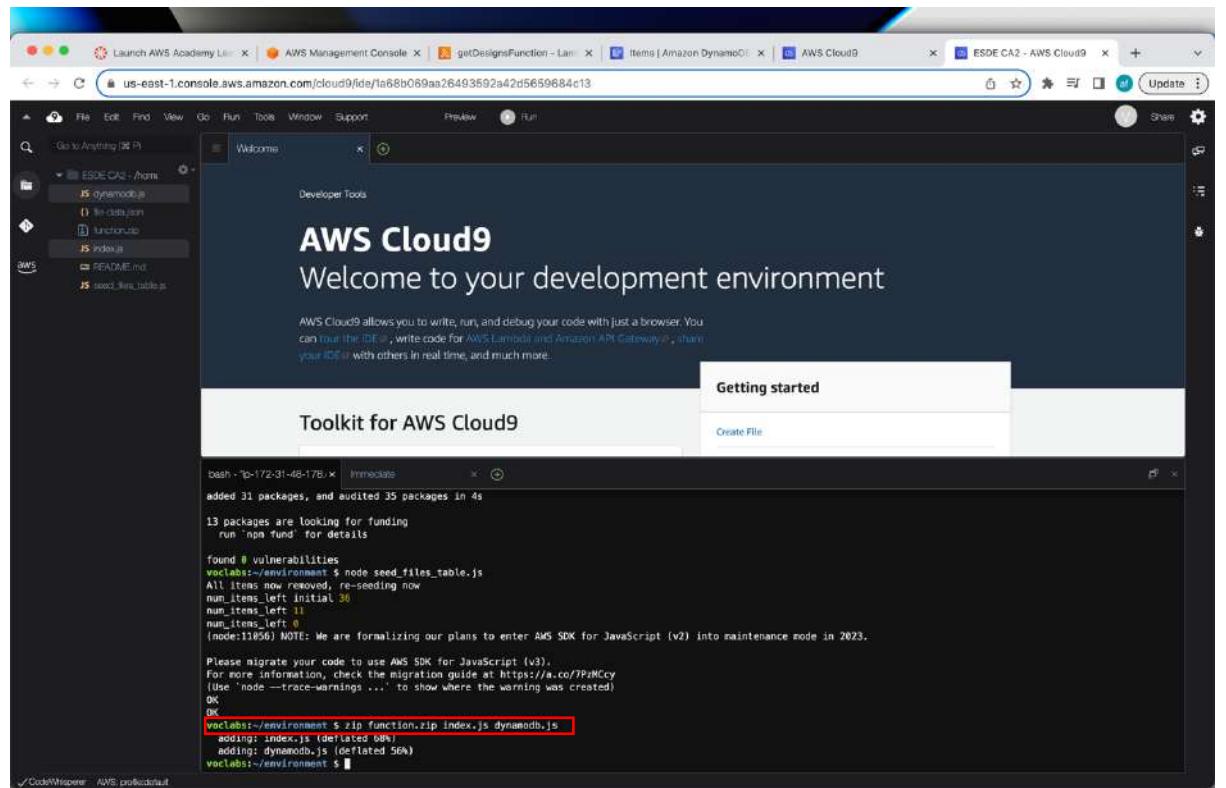
- For the creation of the function, follow the illustration below.



- 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**.



The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file tree with files like `index.js`, `function.json`, `index.html`, and `seed_files_table.js`. The main area has a "Welcome" message and a "Developer Tools" section. Below that is a "Toolkit for AWS Cloud9" panel. A terminal window is open with the following command highlighted in red:

```
bash -lD-172-31-48-17B ~ Immediate
added 31 packages, and audited 35 packages in 4s
13 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
vscLabst:~/environment $ node seed_files_table.js
All items now removed, re-seeding now
num_items_left: initial: 36
num_items_left: 36
num_items_left: 0
num_items_left: 0
(node:11856) NOTE: We are formalizing our plans to enter AWS SDK for JavaScript (v2) into maintenance mode in 2023.
Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the migration guide at https://a.co/TPzMCcy
(use 'node --trace-warnings ...' to show where the warning was created)
OK
OK
vscLabst:~/environment $ zip function.zip index.js dynamodb.js
adding: index.js (deflated 68%)
adding: dynamodb.js (deflated 56%)
vscLabst:~/environment $
```

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

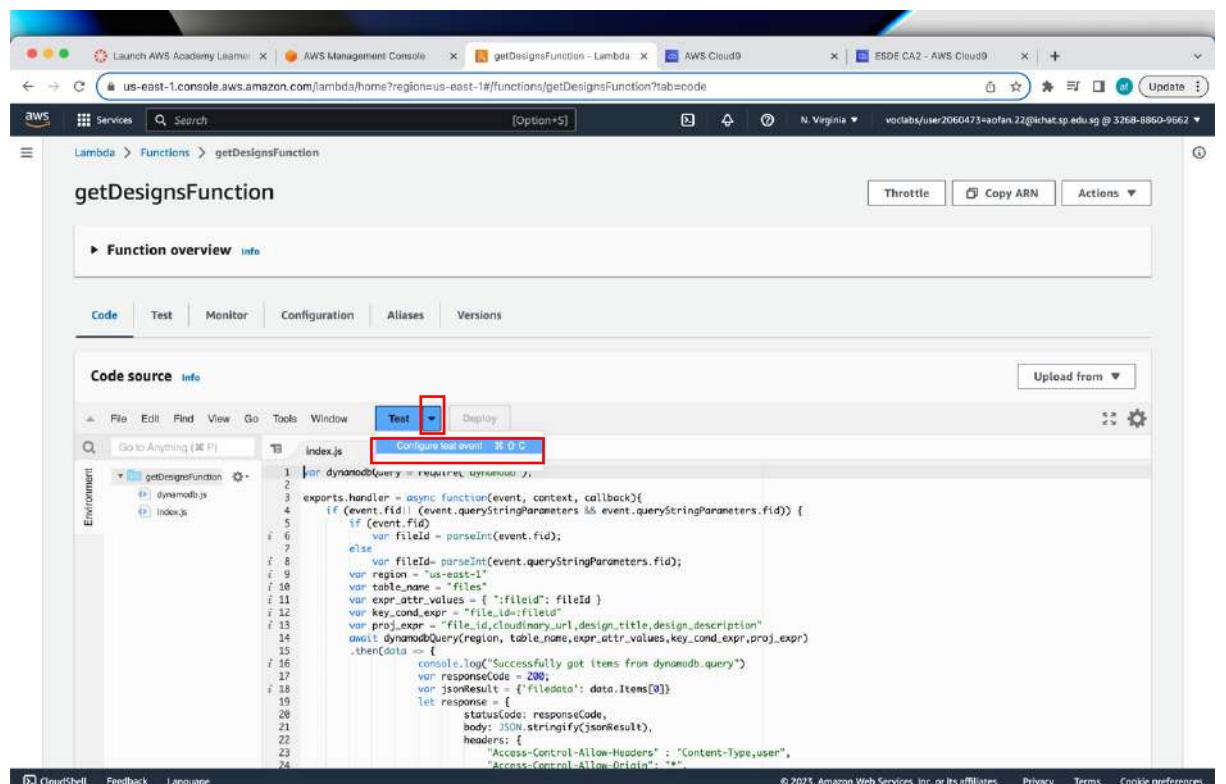
The screenshot shows the AWS Cloud9 developer tools interface. On the left, there's a file tree with files like `index.js`, `bin.database`, `functions`, `README.md`, and `seed_data_table.js`. The main area has a title "AWS Cloud9" and "Welcome to your development environment". Below that is a message about AWS Cloud9 allowing you to write, run, and debug code with just a browser. A terminal window is open with the command `aws lambda update-function-code --function-name getDesignsFunction --zip-file file://function.zip` highlighted in red. The terminal output shows the function configuration details.

```

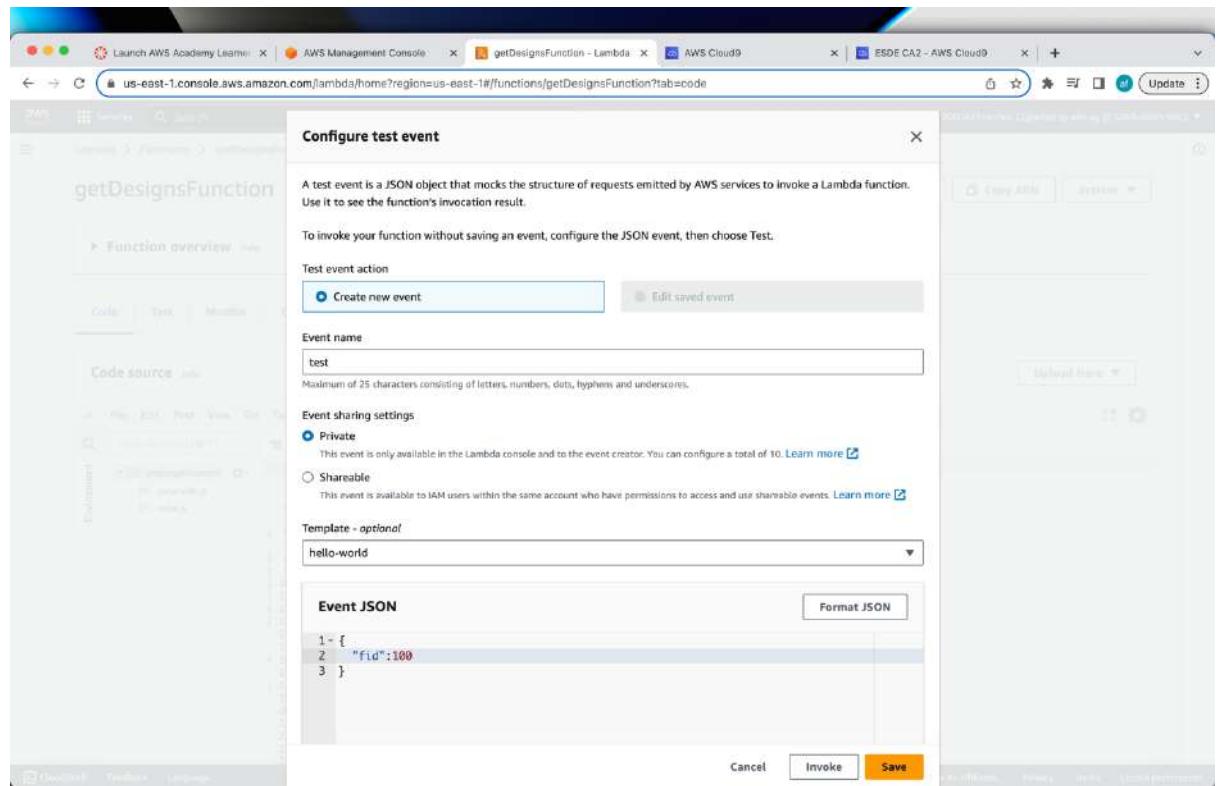
python2 -i p-172-31-48-1 > Immediate
vectabs:~/environment $ aws lambda update-function-code --function-name getDesignsFunction --zip-file file://function.zip
{
    "LastUpdateStatus": "InProgress",
    "FunctionName": "getDesignsFunction",
    "LastModified": "2023-08-13T06:38:20.000+0000",
    "RevisionId": "8cd43e4e-fad1-4b33-be42-1fdd768b674",
    "LastUpdateStatusReason": "The function is being created.",
    "MemorySize": 128,
    "State": "Active",
    "Version": "LATEST",
    "Role": "arn:aws:lambda:us-east-1:326888609662:role/LabRole",
    "Timeout": 3,
    "Handler": "index.handler",
    "Runtime": "nodejs16.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "EsqIHSjY35srnw1VxNt14knAAUqeT7eBTIKKTRbXw",
    "Description": "",
    "LastUpdateStatusReasonCode": "Creating",
    "CodeSize": 1422,
    "FunctionArn": "arn:aws:lambda:us-east-1:326888609662:function:getDesignsFunction",
    "PackageType": "Zip"
}
vectabs:~/environment $ 

```

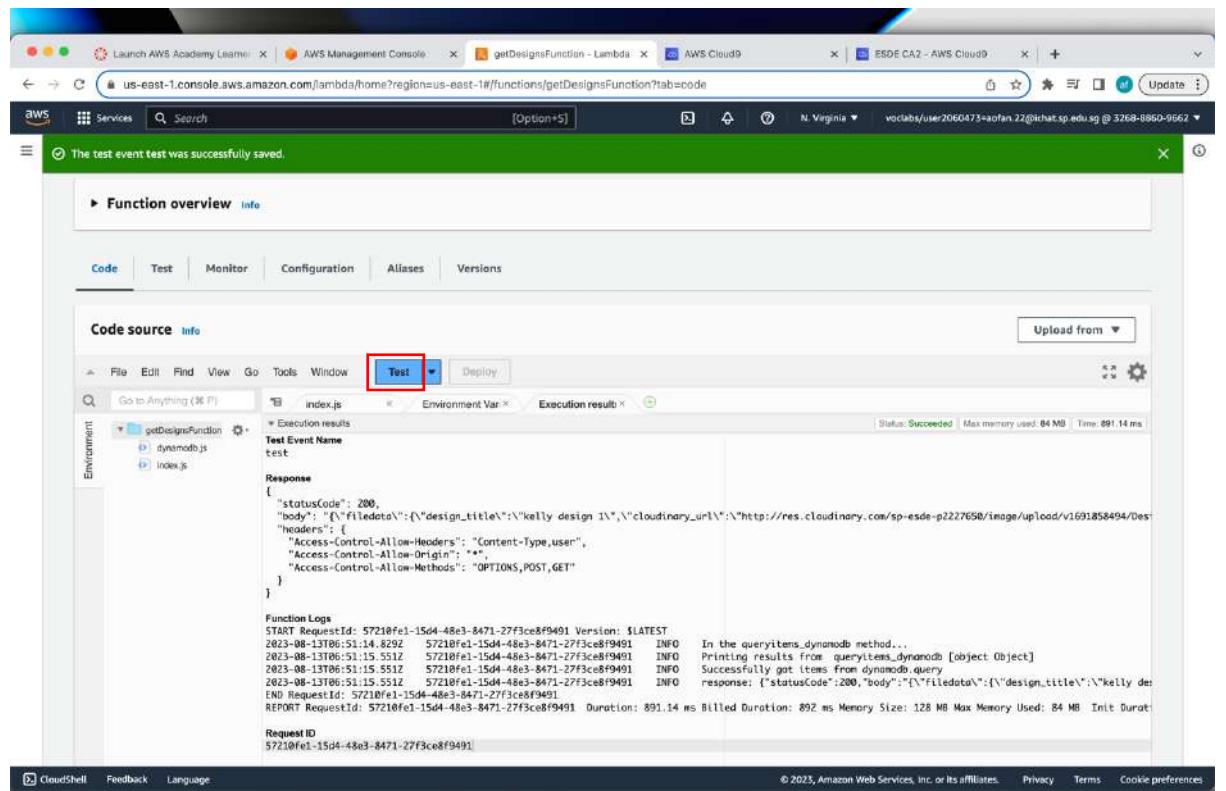
- Go to AWS Lambda > Functions > getDesignsFunction. (the files in the zip file should appear here) Click on the dropdown arrow next to Test and choose Configure test event.



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

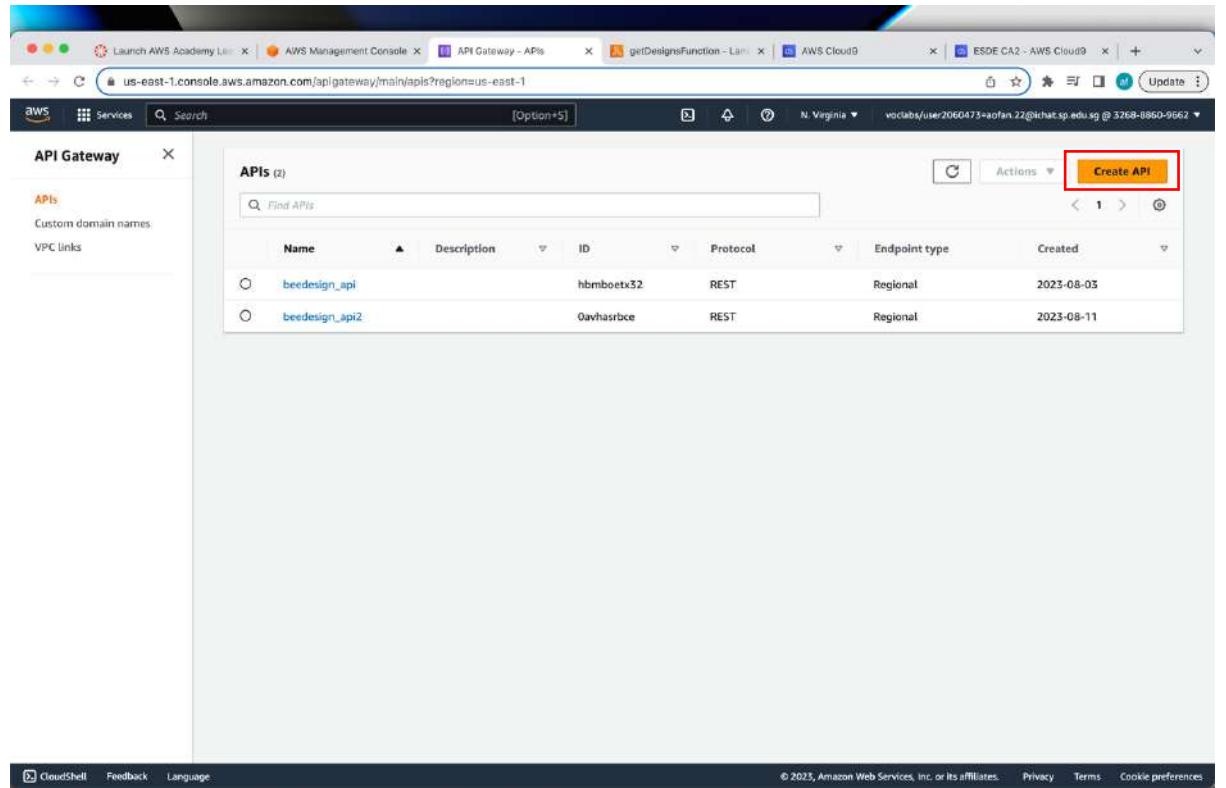


- Hit the Save button and click on Test. If the Lambda function is configured correctly, response similar to the illustration below will be displayed.

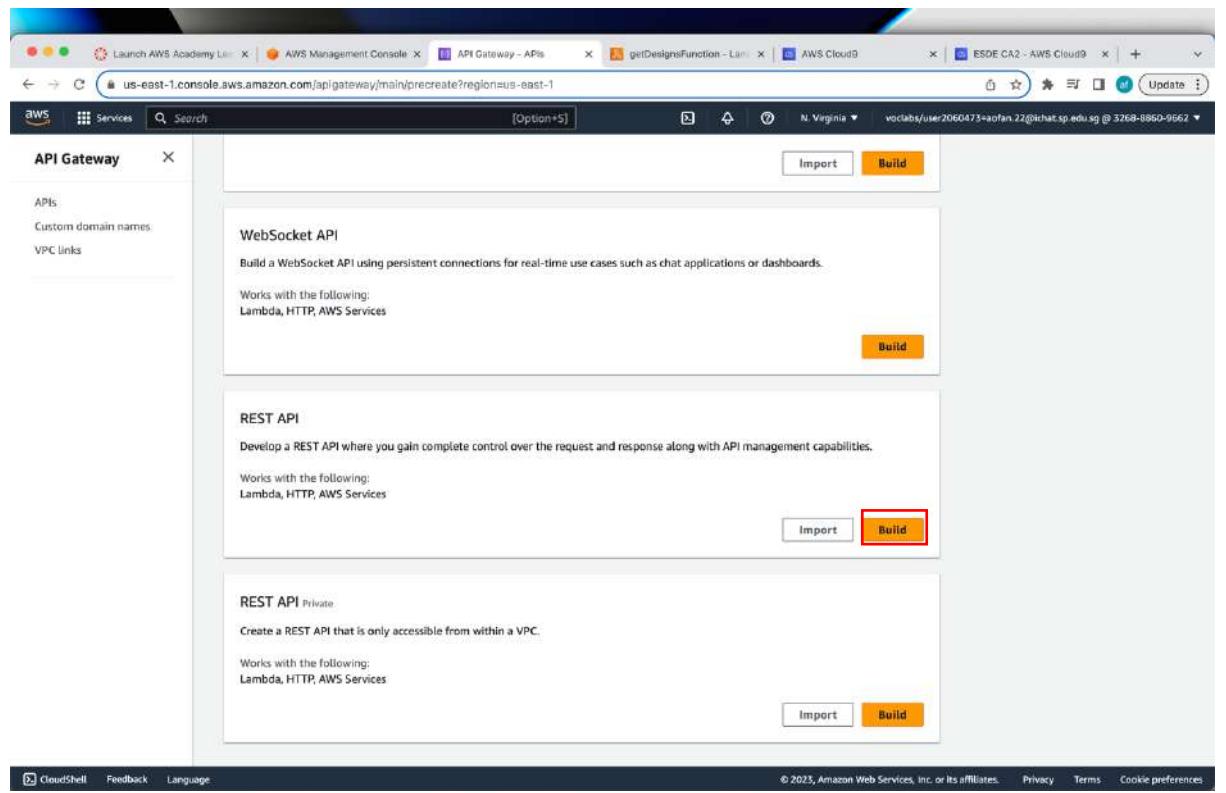


8. Setting up API endpoint for Lambda function using API Gateway

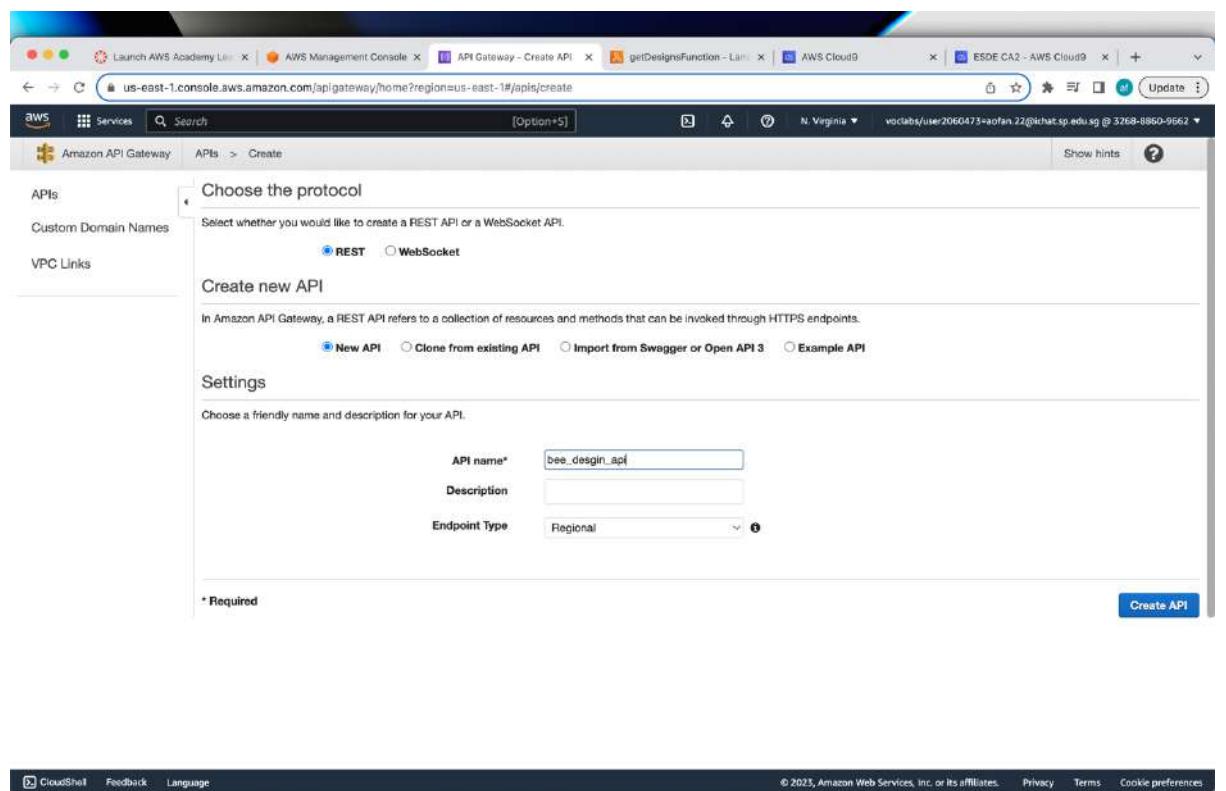
- Navigate to API Gateway and click Create API.



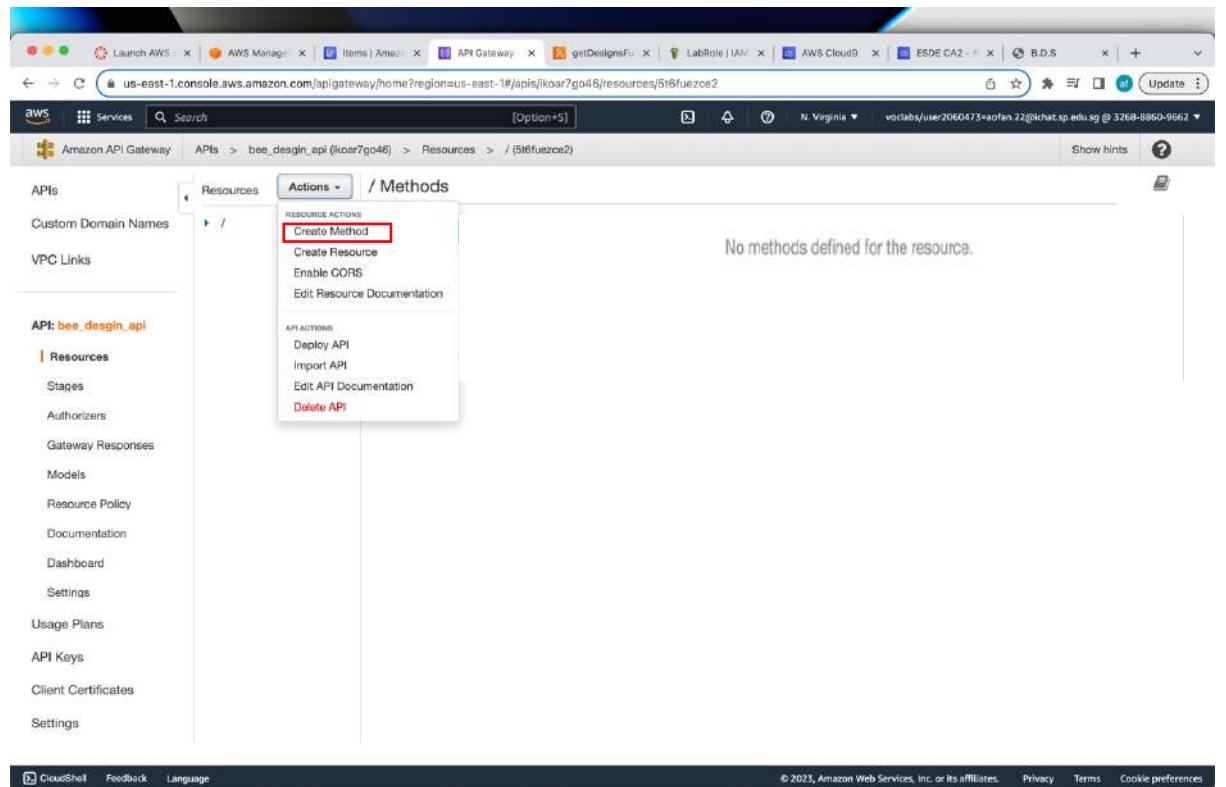
- Choose REST API and click build.



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



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



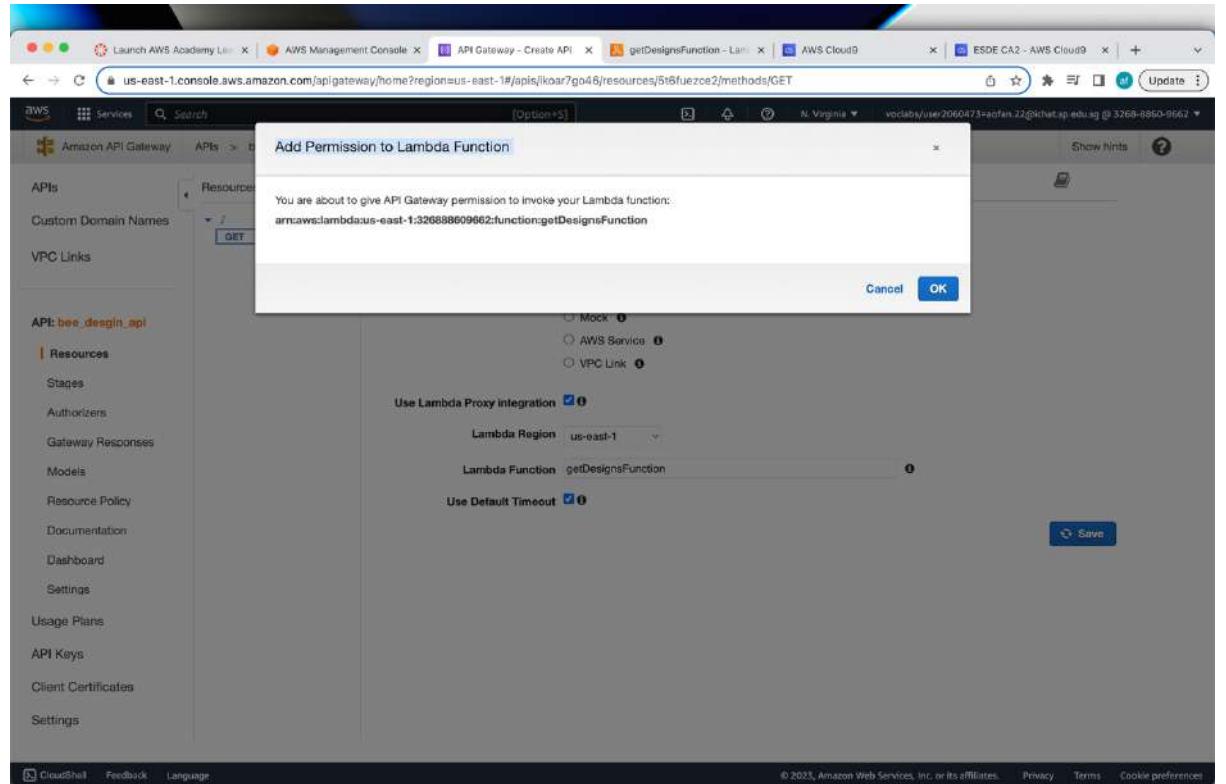
- Click on the dropdown box and select GET. Click on the tick to save.

The screenshot shows the AWS API Gateway console. On the left, a sidebar lists various API resources like Stages, Authorizers, and Models. The main area is titled '/ Methods' and shows a single GET method for the root resource '/'. A red box highlights the 'GET' dropdown menu. To the right, a message states 'No methods defined for the resource.'

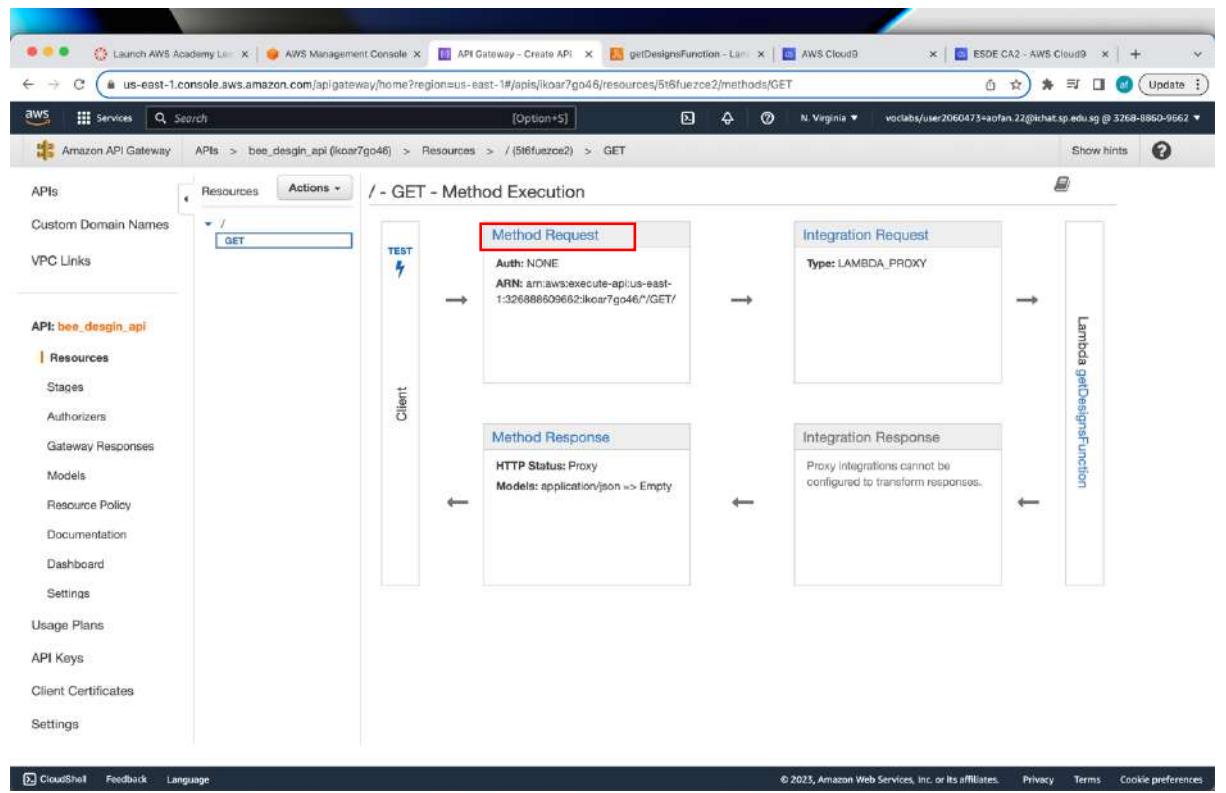
- Set up the method as shown below and click Save.

The screenshot shows the AWS API Gateway console with a new GET method being configured under the resource '/'. The 'Integration type' is set to 'Lambda Function' (radio button selected). The 'Lambda Region' is 'us-east-1' and the 'Lambda Function' is 'getDesignsFunction'. The 'Use Default Timeout' checkbox is checked. A blue 'Save' button is visible at the bottom right.

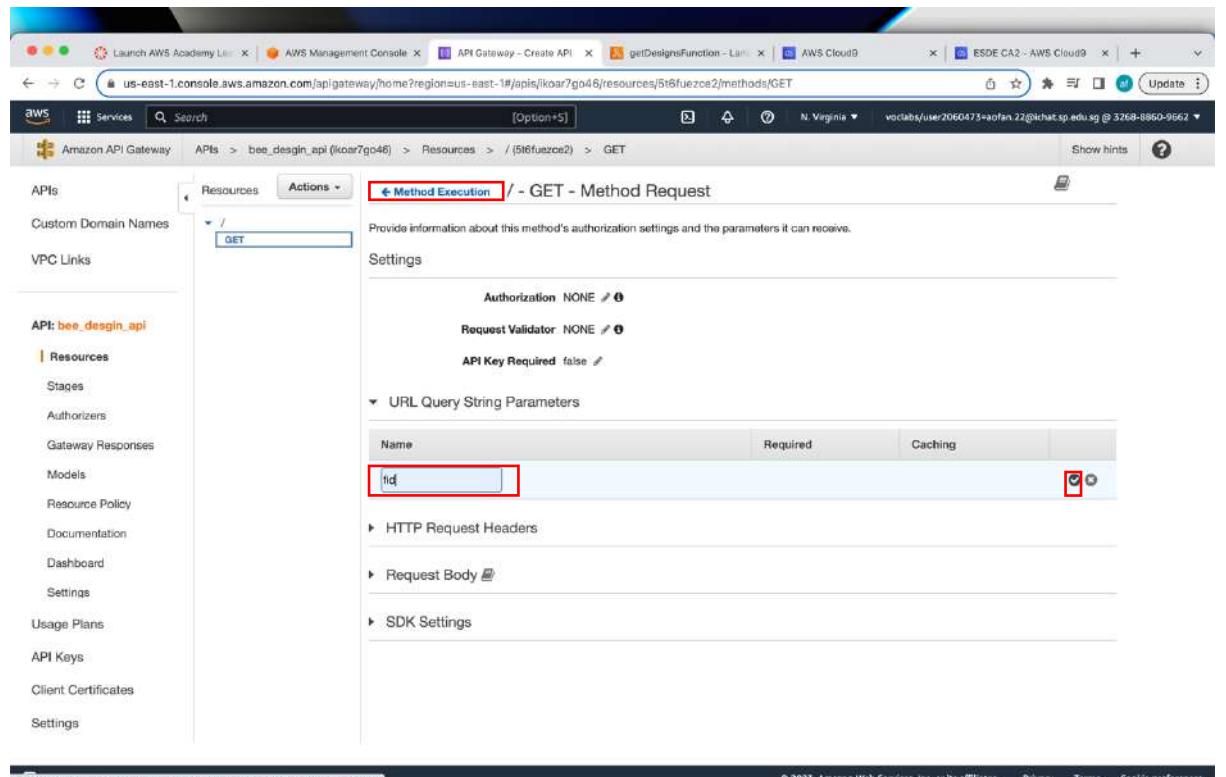
- Click OK when the Add Permission to Lambda Function prompt shows up.



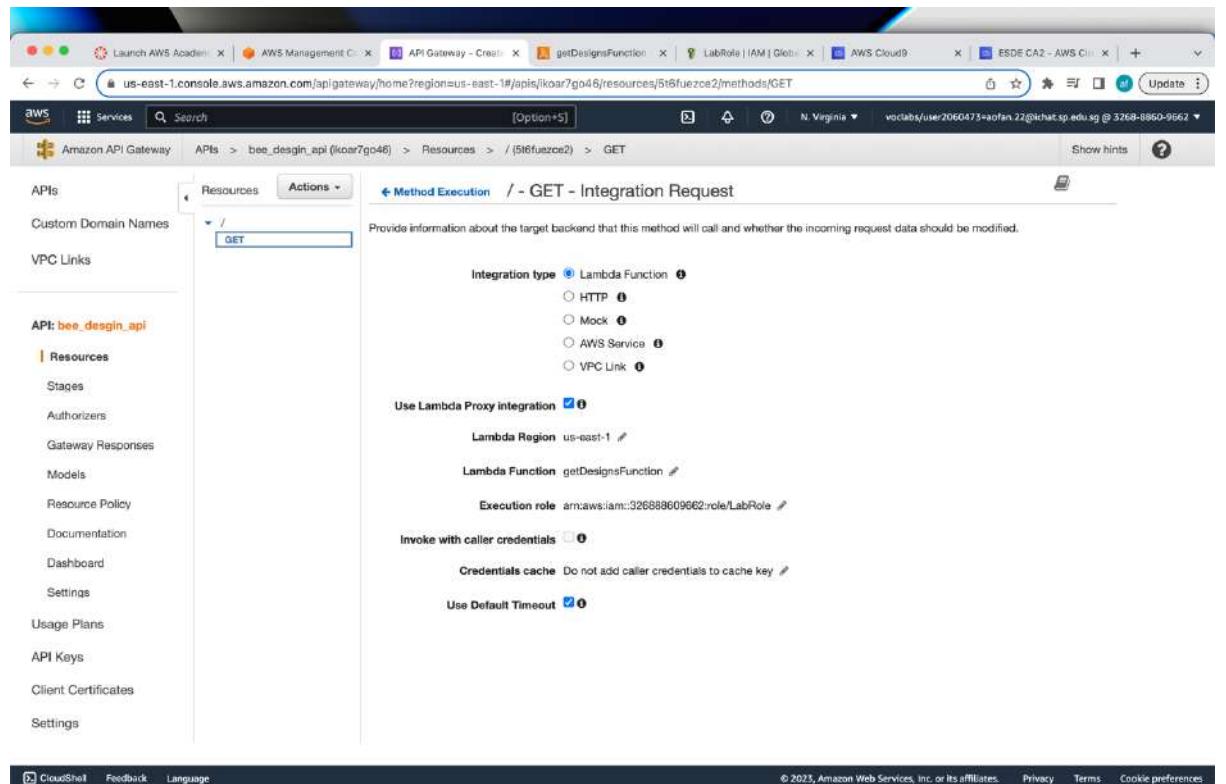
- Select Method Request.



- 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)

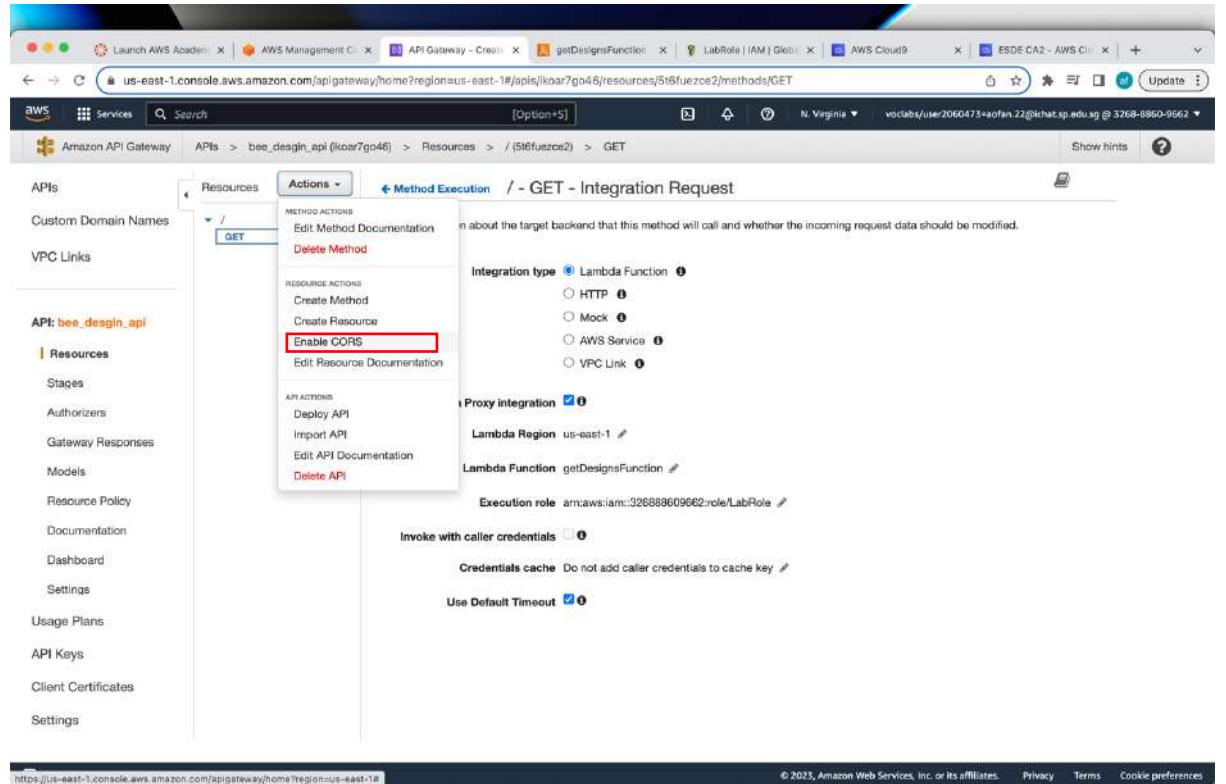


- Go to Lambda and click on LabRole.

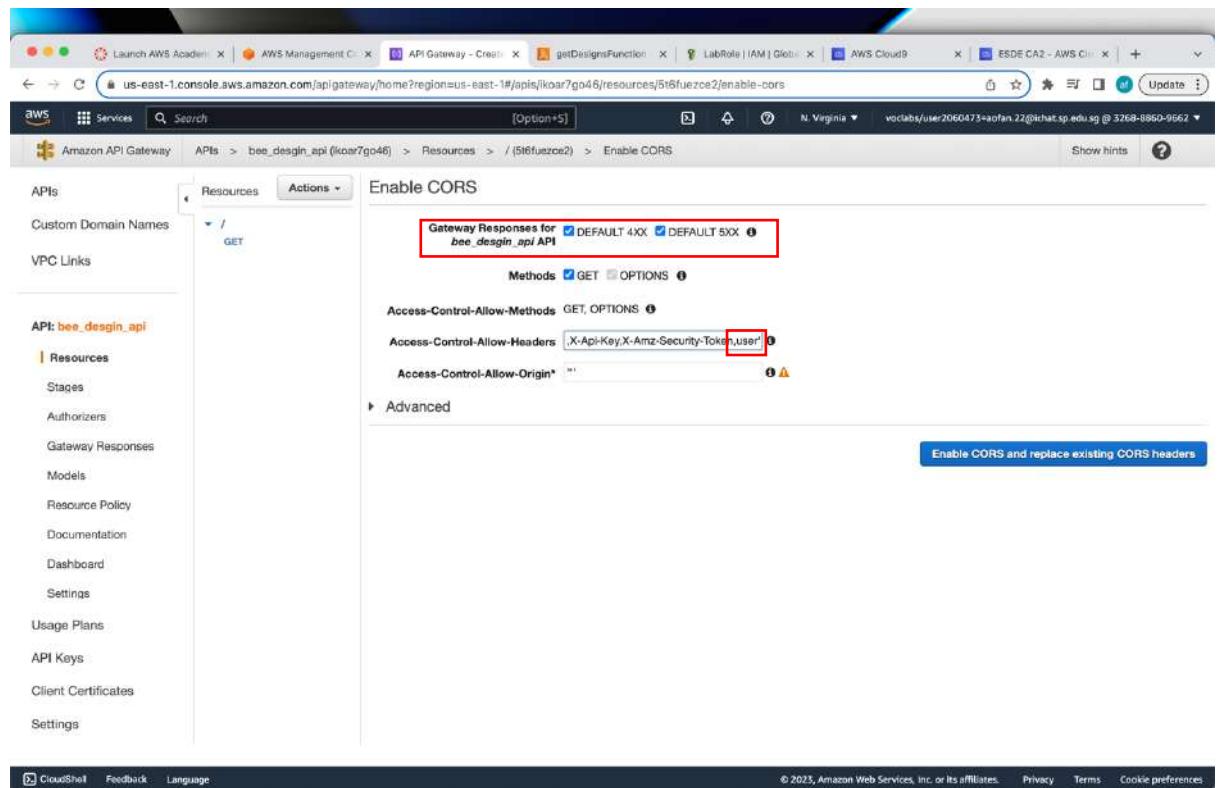
- Copy the ARN and paste it into Execution role.

Policy name	Type	Description
c82452a175248614509714t1w326888609662-VocLabPolicy1-1KJ8FABFIM68X	Customer managed	
c82452a175248614509714t1w326888609662-VocLabPolicy2-16WWPHN9ZOOXZ	Customer managed	
c82452a175248614509714t1w326888609662-VocLabPolicy3-4NU15WE6873T	Customer managed	
AmazonEC2ContainerRegistryReadOnly	AWS managed	Provides read-only access to Amazon ECR

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

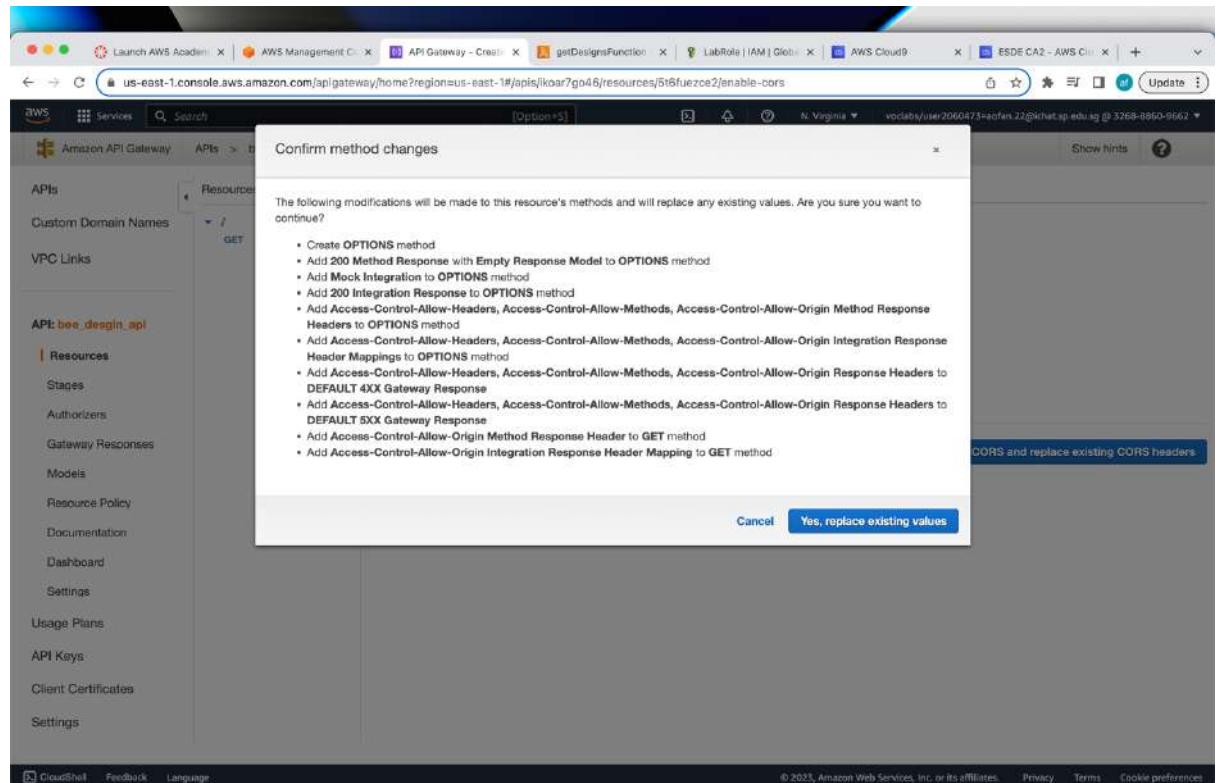


- Check the DEFAULT 4XX and 5XX boxes and add ,**user** in Access-Control-Allow-Headers. Finally, click on Enable CORS and replace existing CROS headers.



The screenshot shows the 'Enable CORS' configuration for a specific API resource. The 'Access-Control-Allow-Headers' field is highlighted with a red box, containing the value 'X-Api-Key,X-Amz-Security-Token,user'. Other fields shown include 'Gateway Responses for bee_desgin_api API' (with checkboxes for 'DEFAULT 4XX' and 'DEFAULT 5XX'), 'Methods' (with 'GET' checked), and 'Access-Control-Allow-Methods' (set to 'GET, OPTIONS'). The 'Advanced' section is collapsed.

- Click Yes, replace existing values to confirm.



The screenshot shows a confirmation dialog titled 'Confirm method changes'. It lists the following modifications:

- Create OPTIONS method
- Add 204 Method Response with Empty Response Model to OPTIONS method
- Add Mock Integration to OPTIONS method
- Add 200 Integration Response to OPTIONS method
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Method Response Headers to OPTIONS method
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Integration Response Header Mappings to OPTIONS method
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Response Headers to DEFAULT 4XX Gateway Response
- Add Access-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Response Headers to DEFAULT 5XX Gateway Response
- Add Access-Control-Allow-Origin Method Response Header to GET method
- Add Access-Control-Allow-Origin Integration Response Header Mapping to GET method

At the bottom right of the dialog is a blue button labeled 'Yes, replace existing values'.

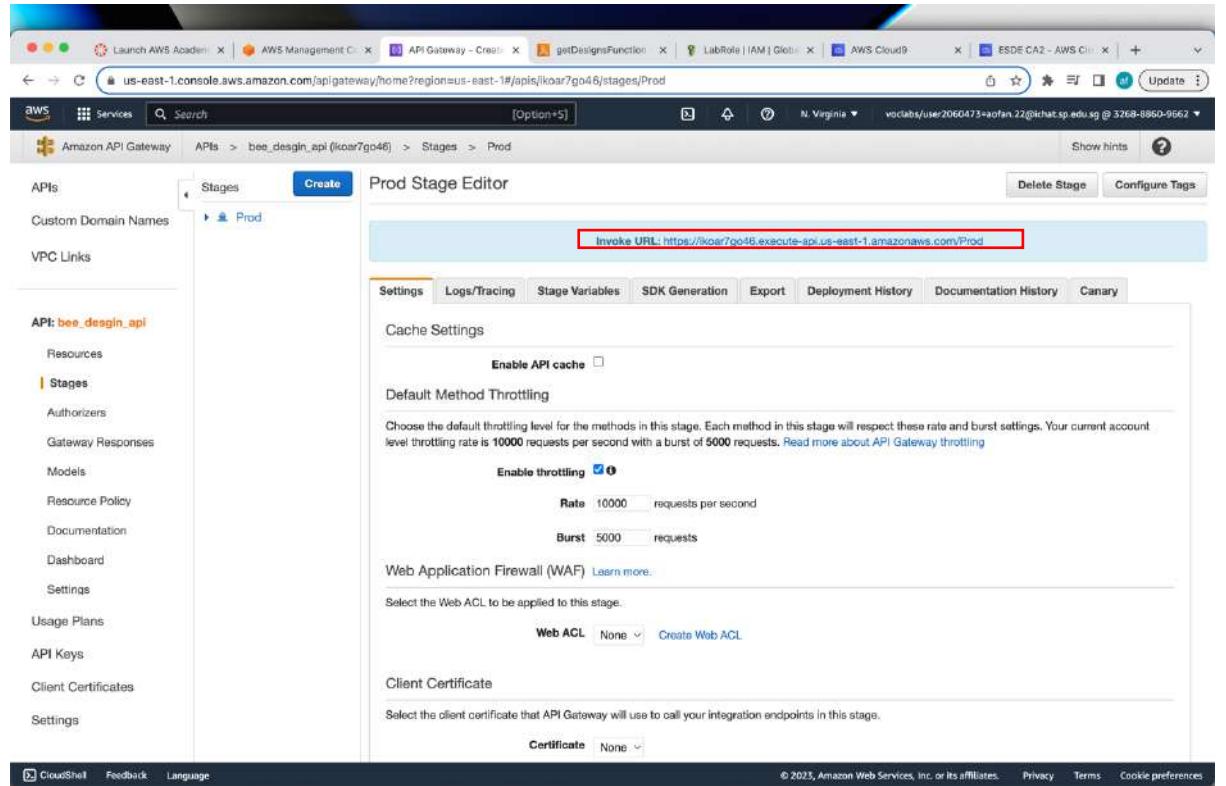
- Click on Actions dropdown button and select Deploy API.

The screenshot shows the AWS API Gateway console. On the left, there's a sidebar with various options like APIs, Custom Domain Names, VPC Links, and the current API named 'bee_desgin_api'. Under 'bee_desgin_api', the 'Resources' section is selected. In the main area, there's a 'Actions' dropdown menu. The 'API ACTIONS' section contains several options: 'Create Method', 'Create Resource', 'Enable CORS', 'Edit Resource Documentation', 'Deploy API' (which is highlighted with a red box), 'Import API', 'Edit API Documentation', and 'Delete API'. Below these, under 'RESOURCE ACTIONS', are 'OPTIONS method', 'Method Response with Empty Response Model to OPTIONS method', 'Integration Response to OPTIONS method', 's-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Method Response Headers to OPTIONS method', 's-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Integration Response Header Mappings to OPTIONS method', 's-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Response Headers to DEFAULT 4XX Gateway', 's-Control-Allow-Headers, Access-Control-Allow-Methods, Access-Control-Allow-Origin Response Headers to DEFAULT 5XX Gateway', 's-Control-Allow-Origin Method Response Header to GET method', and 's-Control-Allow-Origin Integration Response Header Mapping to GET method'. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and 'Language', along with copyright information for Amazon Web Services.

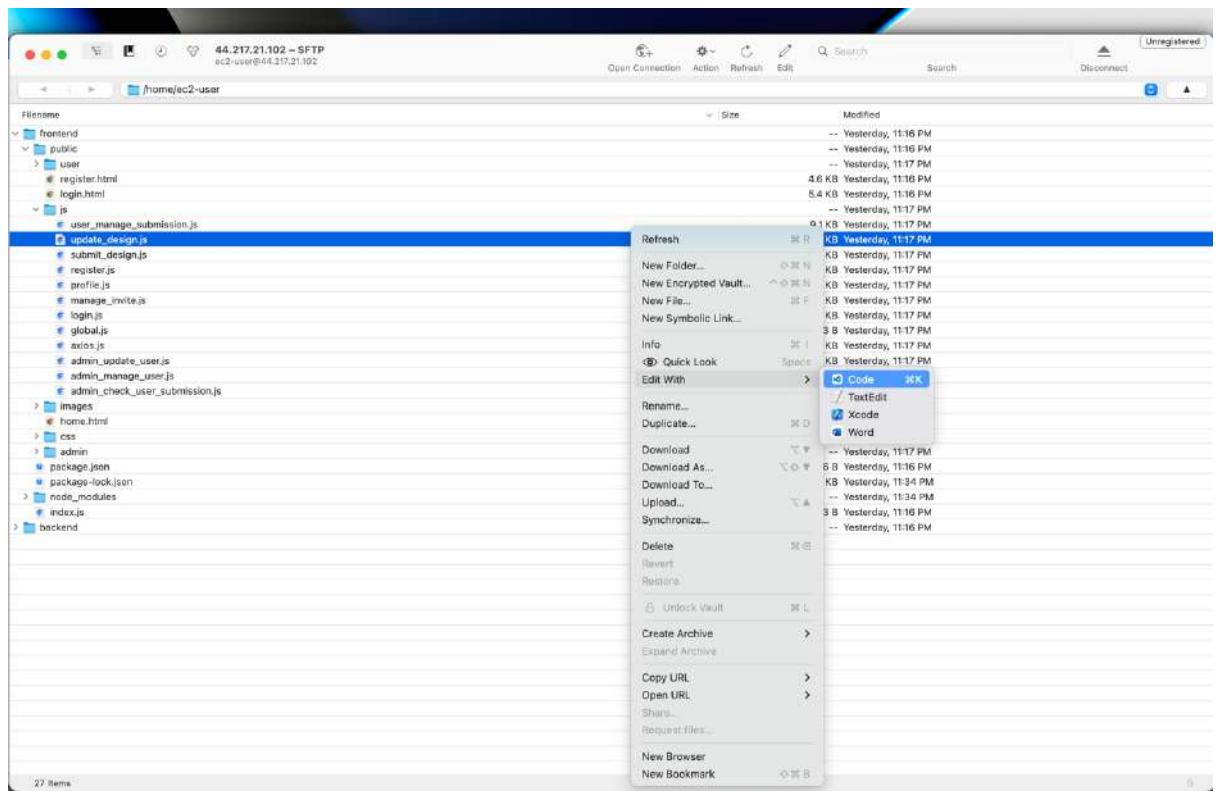
- Set up according to the illustration below and click Deploy.

The screenshot shows the 'Deploy API' dialog box from the AWS API Gateway console. The dialog has a title 'Deploy API' and a sub-instruction '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.' It contains three fields: 'Deployment stage' (set to 'New Stage'), 'Stage name' (set to 'Prod'), and 'Stage description' (empty). Below these fields is a 'Deployment description' input field. At the bottom right of the dialog are 'Cancel' and 'Deploy' buttons. To the right of the dialog, there's a sidebar with deployment-related notes: 's-Allow-Origin Method Response Headers to OPTIONS method', 's-Allow-Origin Integration Response Header Mappings to OPTIONS method', 's-Allow-Origin Response Headers to DEFAULT 4XX Gateway', 's-Allow-Origin Response Headers to DEFAULT 5XX Gateway', and 's-Allow-Origin Response Headers to Prod'. At the very bottom of the page, there are links for 'CloudShell', 'Feedback', and 'Language', along with copyright information for Amazon Web Services.

- 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)

```
private var folders = 63; //279mgs39938_cv0ax35m000gn T: ch.usd0.cybanduck > editor-86d7d2-e5b5-490c-8058-efb603103a09 > home > ec2-user > frontend > public > [a > -866861645 > update_design.js > getOldData.js] > update_design.js

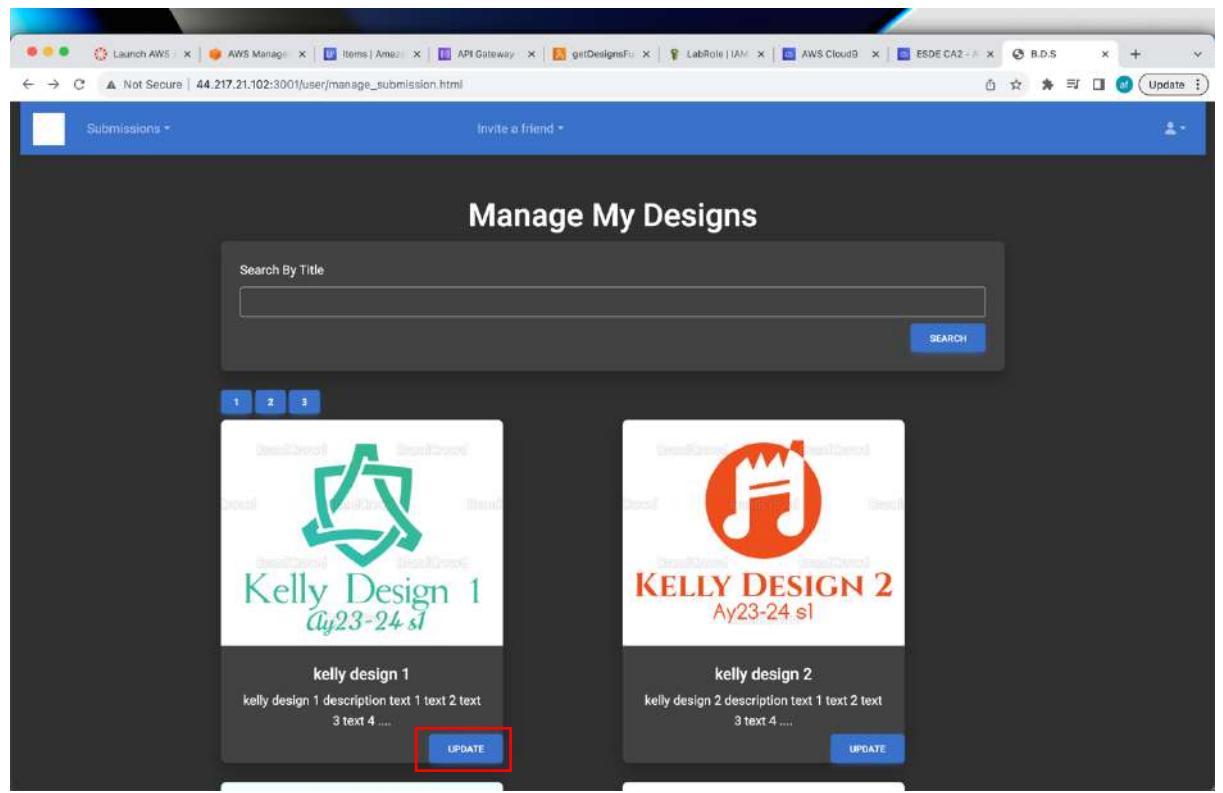
44      );
45    });
46  });
47  $('#basicButton1').on("click", function(e) {
48    e.preventDefault();
49    window.history.back();
50  });
51
52
53  function getOldData() {
54    const headers = { "Content-Type": "application/x-www-form-urlencoded" };
55    const url = "https://lambda-test.s3.amazonaws.com/Prod7f1da";
56    let query = window.location.search.substring(1);
57    let arrayData = query.split("&");
58    let fileId = arrayData[0];
59    console.dir('Obtained file id from URL : ', fileId);
60    let userId = localStorage.getItem('user_id');
61    axios({
62      headers: {
63        'Content-Type': 'application/x-www-form-urlencoded',
64        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4285.147 Safari/537.36',
65        'Accept': 'application/json'
66      },
67      method: 'get',
68      url: 'https://lambda-test.s3.amazonaws.com/Prod7f1da' + fileId,
69    })
70    .then(function(response) {
71      //Using the following to inspect the response.data data structure
72      //Before doing the code which dynamically populate the elements with data:
73      //console.dir(response.data);
74      const record = response.data;
75      const recordId = record.id;
76      $( "#fileIdInput" ).val(record.file_id);
77      $( "#designDescriptionInput" ).val(record.design_description).focus();
78
79      $( "#designImage" ).attr('src', record.cloudinary_url).focus();
80    })
81    .catch(function(response) {
82      //Handle error
83      console.dir(response);
84      if (response.status === 404) {
85        type: "error",
86        timeout: "6000",
87        layout: "topCenter",
88        theme: "sunset",
89        text: "Unable retrieve file data",
90      }).show();
91    }
92  }
93
94  //End of getOldData
95  //Call getOldData function to do a GET HTTP request on an API to retrieve one user record
96  getOldData(); //Call getOldData to begin populating the form input controls with the existing record information.
97
98 } //End of checking for updateDesignFormContainer()Query object
```

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

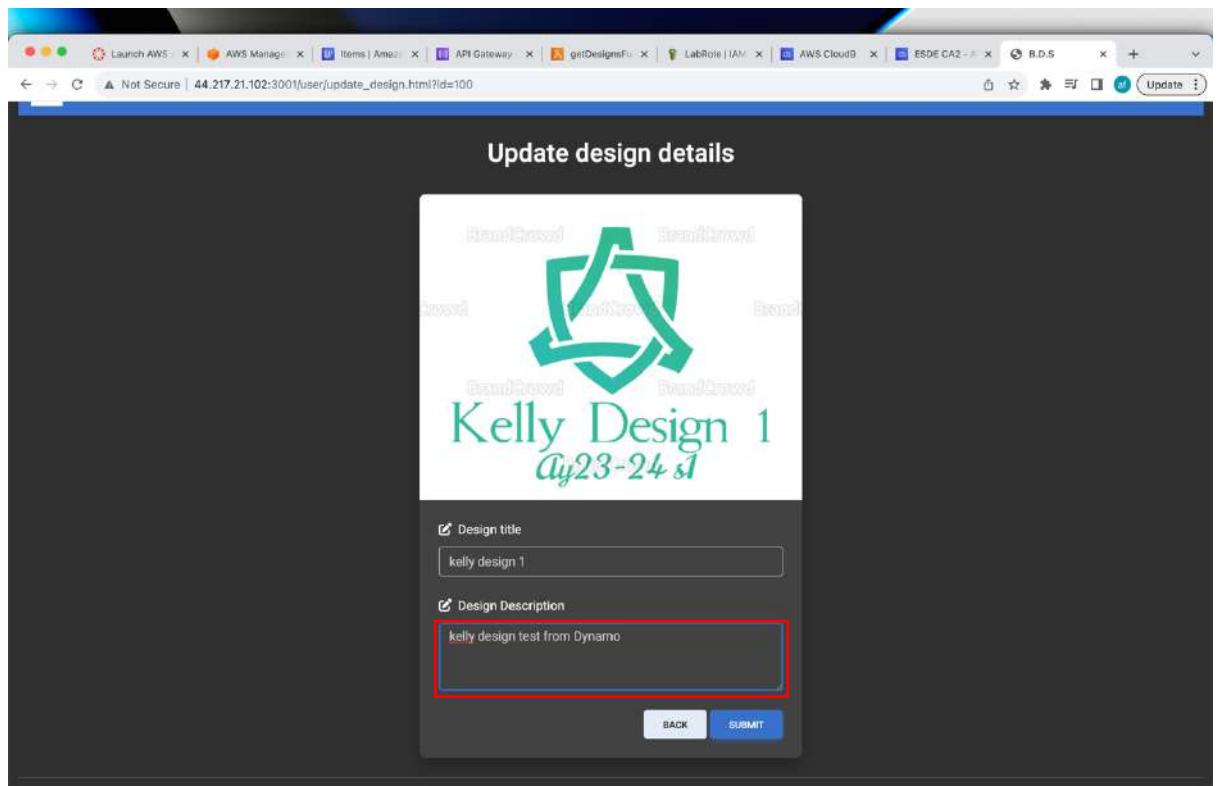
The screenshot shows the AWS DynamoDB console with the 'files' table selected. A modal dialog is open, allowing the user to edit the 'design_description' field for a specific item. The modal contains a text input field with the value '100' highlighted. A red box surrounds the modal and the 'Save' button, indicating the action to be taken.

ary_file_id	cloudinary_url	created_by_id	design_description	design_title
ftg4whvahdrv...	http://res.cloudin...	100	kelly design 1	kelly design 1
butvfwblfrijey...	http://res.cloudin...	100	kelly design 2	kelly design 2
skviasfdb8i1g...	http://res.cloudin...	100	kelly design 3	kelly design 3
elat9tcwo1twc...	http://res.cloudin...	100	kelly design 4 descripti...	kelly design 4
updnpi19kzsry...	http://res.cloudin...	100	kelly design 5 descripti...	kelly design 5
cvevoywc0sko...	http://res.cloudin...	100	kelly design 6 descripti...	kelly design 6
bg3kbzseigjgb...	http://res.cloudin...	100	kelly design 7 descripti...	kelly design 7

- 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)

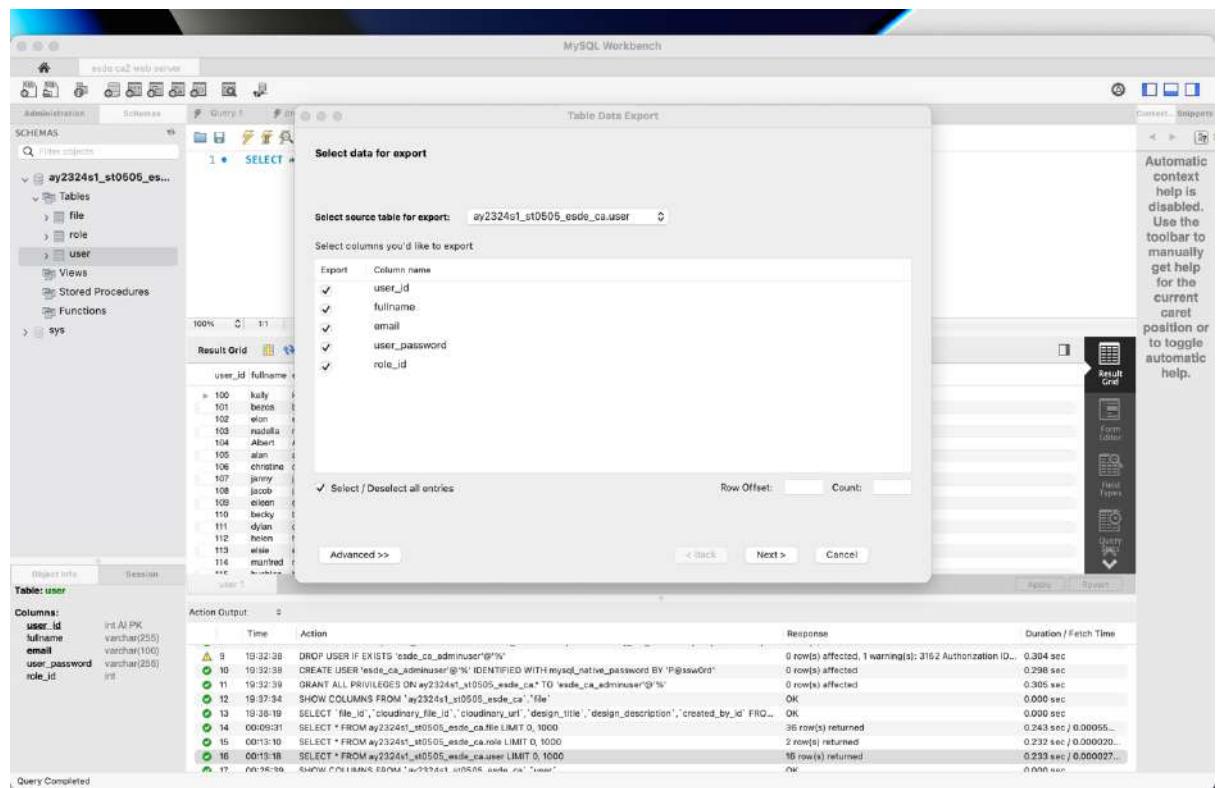


- If you see the edited design_description, it means that the API endpoint is functional.

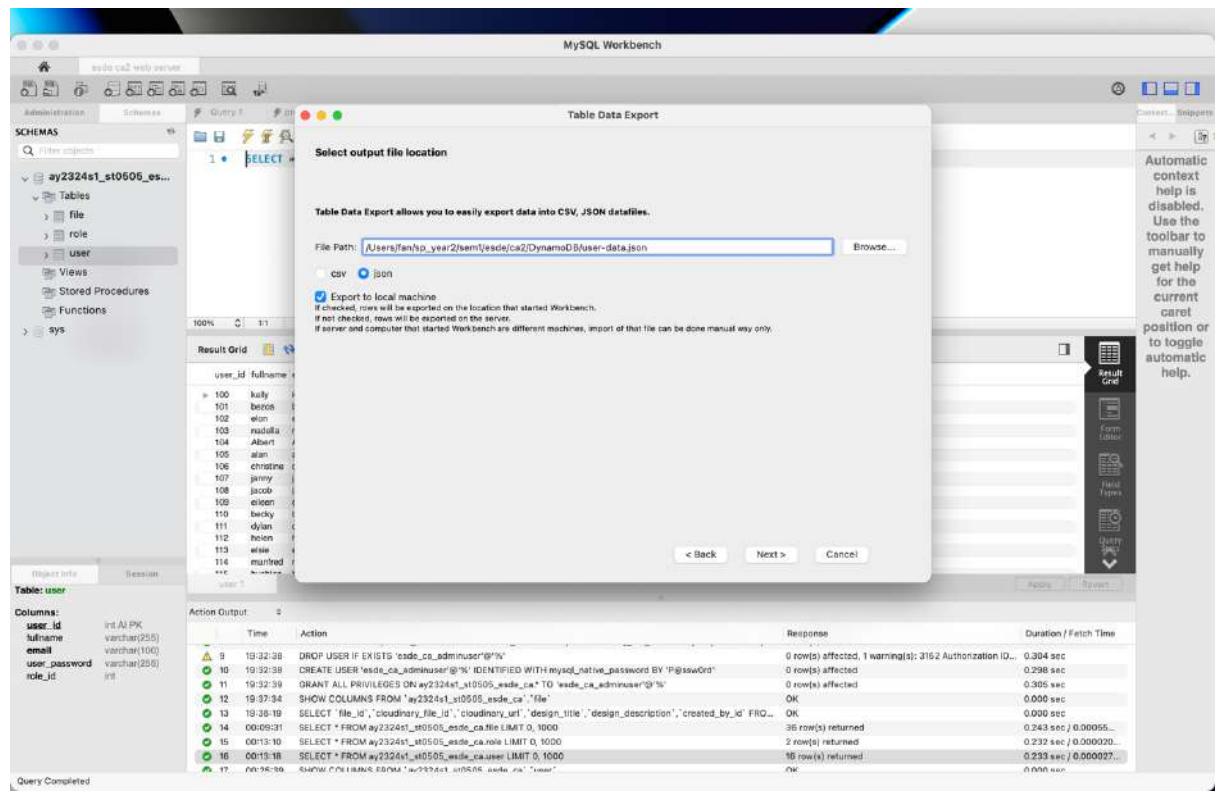


9. Additional Features

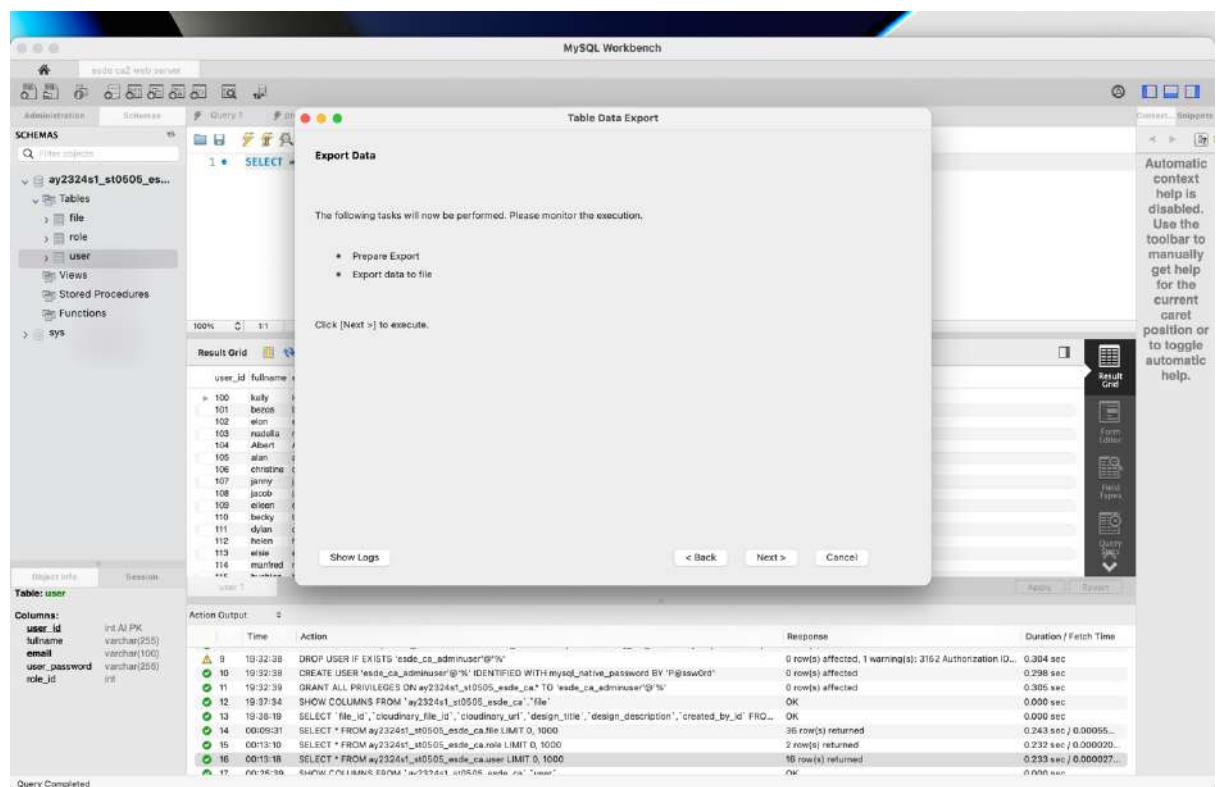
- Go to MySQLWorkbench, click on the esde ca2 web server connection. Right click on user located under ay2324s1_st0505_esde_ca, under Tables and select Data Table Export Wizard.
- On Table Data Export tab, click Next >.



- 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.

Result Grid

user_id	fullname
100	kathy
101	beross
102	elon
103	natalia
104	Albert
105	alan
106	christine
107	jenny
108	jacob
109	eileen
110	becky
111	dylan
112	heidi
113	erica
114	marilyn

Show Logs

< Back Next > Cancel

Action Output

Time	Action	Response	Duration / Fetch Time
9 19:32:38	DROP USER IF EXISTS 'esde_ca_adminuser'@'%'	0 rows affected, 1 warning(s); 3162 Authorization ID...	0.304 sec
10 19:32:39	CREATE USER 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd'	0 row(s) affected	0.298 sec
11 19:32:39	GRANT ALL PRIVILEGES ON ay2324s1_st0505_esde_ca TO 'esde_ca_adminuser'@'%'	0 row(s) affected	0.305 sec
12 19:37:34	SHOW COLUMNS FROM 'ay2324s1_st0505_esde_ca`.`file`	OK	0.000 sec
13 19:38:19	SELECT * FROM ay2324s1_st0505_esde_ca_file LIMIT 0, 1000	OK	0.000 sec
14 00:09:31	SELECT * FROM ay2324s1_st0505_esde_ca_file LIMIT 0, 1000	36 row(s) returned	0.243 sec / 0.00055...
15 00:13:10	SELECT * FROM ay2324s1_st0505_esde_ca_role LIMIT 0, 1000	2 row(s) returned	0.232 sec / 0.000020...
16 00:19:18	SELECT * FROM ay2324s1_st0505_esde_ca_cause LIMIT 0, 1000	16 row(s) returned	0.233 sec / 0.000027...
17 00:35:39	SHOW CREATE TABLE 'ay2324s1_st0505_esde_ca`.`user`	OK	0.000 sec

Query Completed

MySQL Workbench

Table Data Export

Export Results

File /Users/fan/sp_year2/semt/esde/ca/DynamoDB/user-data.json was exported in 0.479 s

Exported 16 records

Result Grid

user_id	fullname
100	kathy
101	beross
102	elon
103	natalia
104	Albert
105	alan
106	christine
107	jenny
108	jacob
109	eileen
110	becky
111	dylan
112	heidi
113	erica
114	marilyn

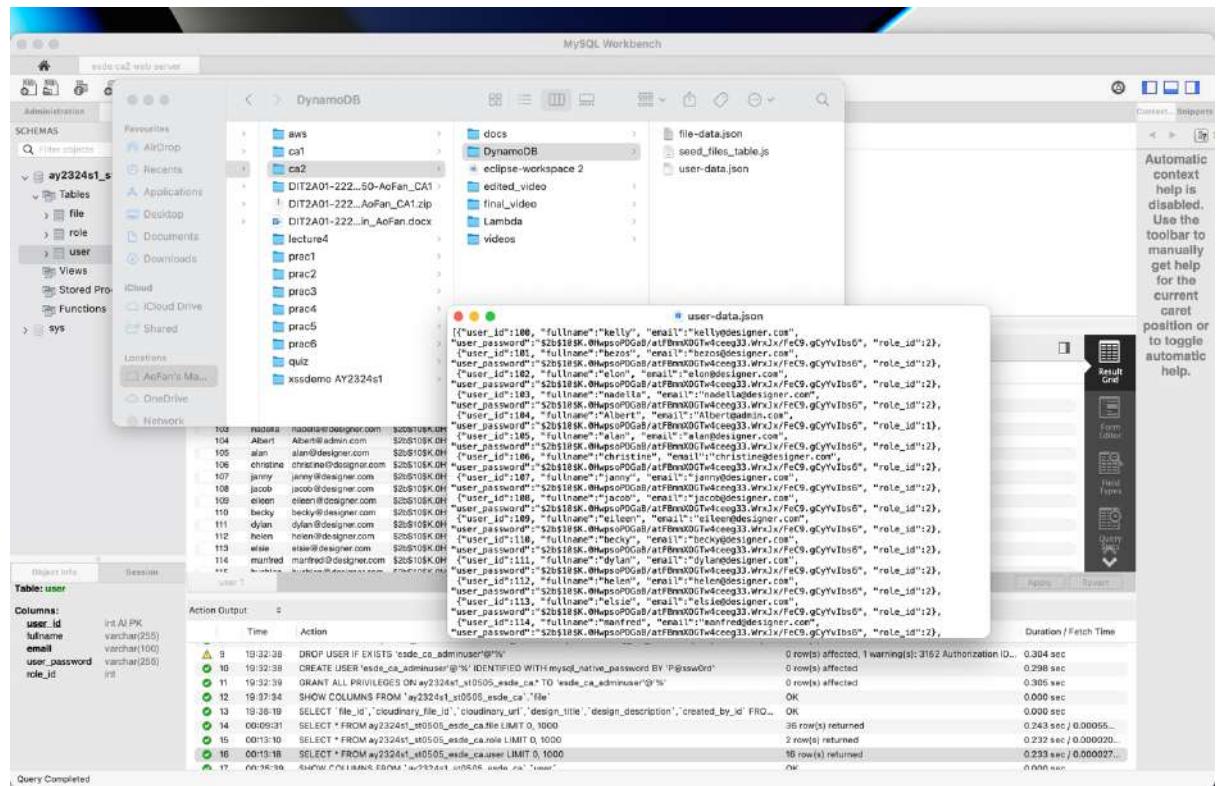
< Back Finish Cancel

Action Output

Time	Action	Response	Duration / Fetch Time
9 19:32:38	DROP USER IF EXISTS 'esde_ca_adminuser'@'%'	0 rows affected, 1 warning(s); 3162 Authorization ID...	0.304 sec
10 19:32:39	CREATE USER 'esde_ca_adminuser'@'%' IDENTIFIED WITH mysql_native_password BY 'P@ssw0rd'	0 row(s) affected	0.298 sec
11 19:32:39	GRANT ALL PRIVILEGES ON ay2324s1_st0505_esde_ca TO 'esde_ca_adminuser'@'%'	0 row(s) affected	0.305 sec
12 19:37:34	SHOW COLUMNS FROM 'ay2324s1_st0505_esde_ca`.`file`	OK	0.000 sec
13 19:38:19	SELECT * FROM ay2324s1_st0505_esde_ca_file LIMIT 0, 1000	OK	0.000 sec
14 00:09:31	SELECT * FROM ay2324s1_st0505_esde_ca_file LIMIT 0, 1000	36 row(s) returned	0.243 sec / 0.00055...
15 00:13:10	SELECT * FROM ay2324s1_st0505_esde_ca_role LIMIT 0, 1000	2 row(s) returned	0.232 sec / 0.000020...
16 00:19:18	SELECT * FROM ay2324s1_st0505_esde_ca_cause LIMIT 0, 1000	16 row(s) returned	0.233 sec / 0.000027...
17 00:35:39	SHOW CREATE TABLE 'ay2324s1_st0505_esde_ca`.`user`	OK	0.000 sec

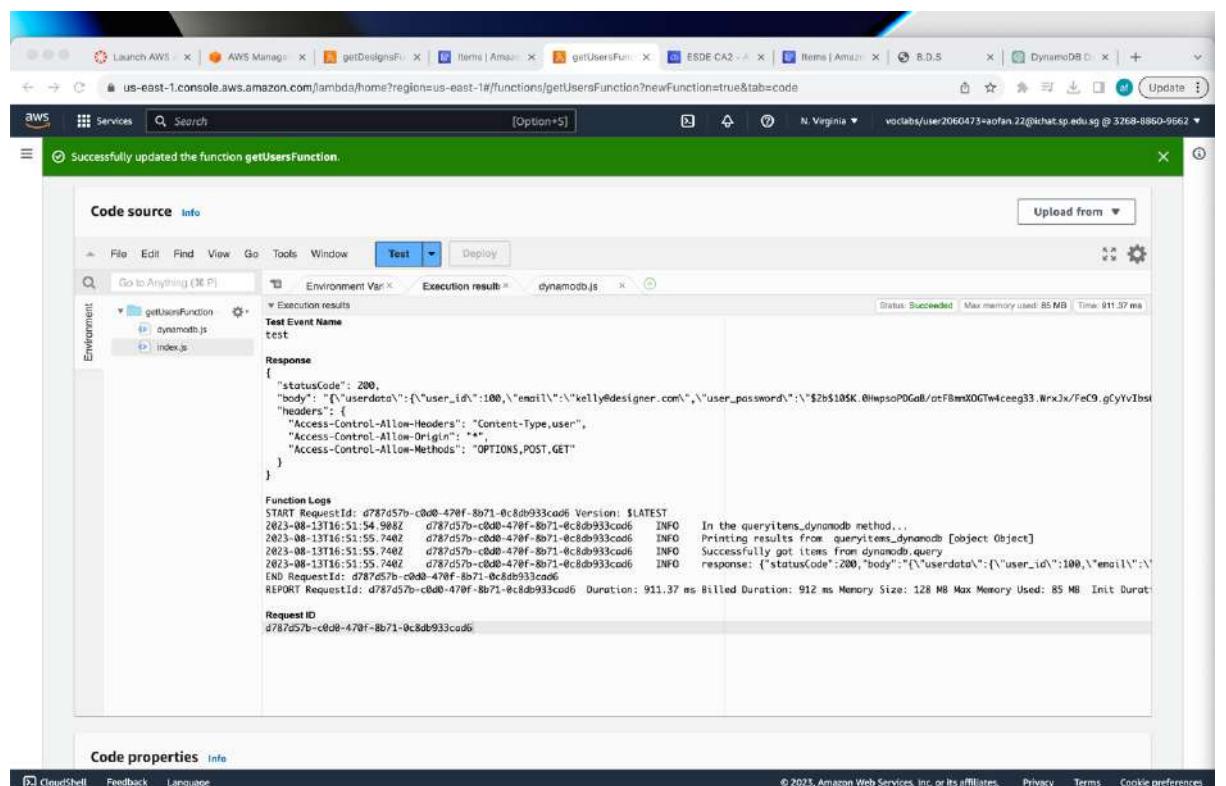
Query Completed

- The file exported should look like this inTextEdit.



- 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.



The screenshot shows the AWS Lambda Test interface. At the top, a green banner says "Successfully updated the function getUsersFunction." Below it, the "Code source" tab is selected. The "Test" button is highlighted. The "Execution result" section shows the following output:

```

Environment Var X Execution result: dynamodb.js
Test Event Name: test
Response:
{
  "statusCode": 200,
  "body": "{\"userdata\":{\"user_id\":100,\"email\":\"kelly@designer.com\",\"user_password\":\"$2b$10$K.0ImpsoP0GqB/ctFBmnX0GTw4ceeg33.WrxJx/Fc9.gCyVvIbsI\"}},
  "headers": {
    "Access-Control-Allow-Headers": "Content-Type,User",
    "Access-Control-Allow-Origin": "*",
    "Access-Control-Allow-Methods": "OPTIONS,POST,GET"
  }
}

Function Logs
START RequestId: d787d57b-c0d0-470f-8b71-0c8db933cad6 Version: $LATEST
2023-08-13T16:51:54.988Z  d787d57b-c0d0-470f-8b71-0c8db933cad6  INFO  In the queryitems_dynamodb method...
2023-08-13T16:51:55.740Z  d787d57b-c0d0-470f-8b71-0c8db933cad6  INFO  Printing results from queryitems_dynamodb [object Object]
2023-08-13T16:51:55.740Z  d787d57b-c0d0-470f-8b71-0c8db933cad6  INFO  Successfully got items from dynamodb.query
2023-08-13T16:51:55.740Z  d787d57b-c0d0-470f-8b71-0c8db933cad6  INFO  response: {"statusCode":200,"body": "{\"userdata\":{\"user_id\":100,\"email\":\"kelly@designer.com\"}}}
END RequestId: d787d57b-c0d0-470f-8b71-0c8db933cad6 Duration: 911.37 ms Billed Duration: 912 ms Memory Size: 128 MB Max Memory Used: 85 MB Init Durat: 0 ms
Request ID: d787d57b-c0d0-470f-8b71-0c8db933cad6

```

At the bottom, there are tabs for "Code properties" and links to "CloudShell", "Feedback", and "Language".

- 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)