**API DOCUMENTATION**

**EC2 URL -> http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com/airlines**

1. **Airlines CRUD endpoint**
2. **Search Airlines by country code**

* HTTP METHOD: **GET**
* URL -> **/airlines?country=<country\_code>**
* Description**:** Returns a list of airlines based on the specified country code
* Input Parameters: country a two-letter country code (e.g., BQ, AW, AG)
* **Output Format:**

[

{

"name": "Leeward Islands Air Transport"

},

{

"name": "Tiara Air"

}

]

- **Status Codes**:

1. 200: Success
2. 400: Missing country parameter
3. 404: No airlines found for the specified country codes ${ list of countries }
4. 500: Internal server error

Test Endpoint ->

1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?country=bq,aw,ag**

Expected output ->

[

{

"name": "Leeward Islands Air Transport"

},

{

"name": "Tiara Air"

}

]

1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?country**

Expected output ->

{

"error": "Missing country parameter"

}

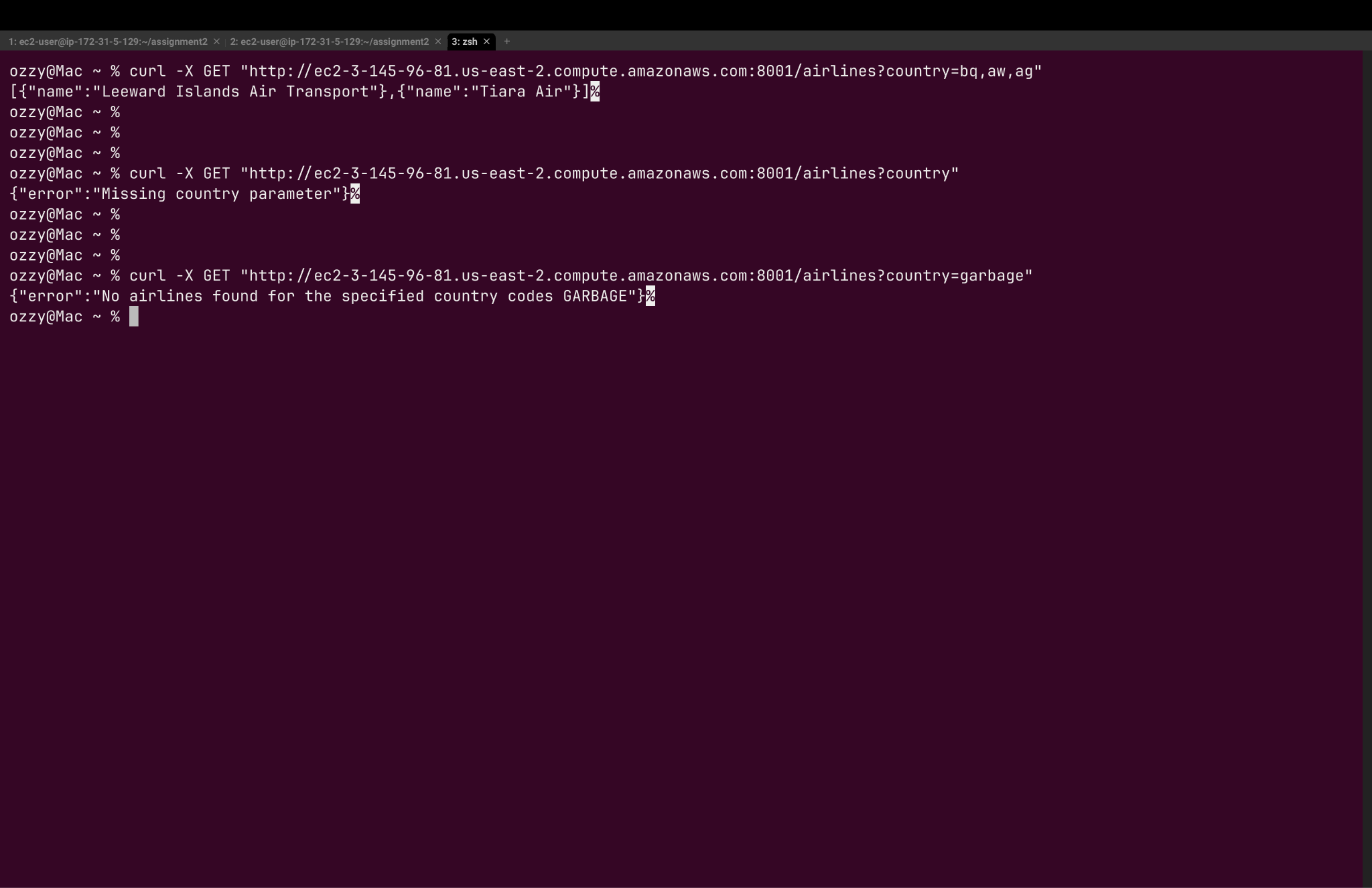
* 1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?country=garbage**

Expected output ->

{

"error": "No airlines found for the specified country codes GARBAGE"

}

Curl Image -> 

**B) Search Airline by ICAO/IATA Code**

* HTTP METHOD: **GET**
* URL -> **/airline?iata=<iata code>&icao=<icao code>**
* Description - return an airline based on icao and/or iata codes
* Input Parameters: A 3-letter ICAO code and or 2-letter IATA code
* **Output Format:**

[

{

"name": "1Time Airline"

}

]

- **status Codes**:

1. 200: Success
2. 400: Please provide at least one of ICAO or IATA
3. 404: No airlines found with the provided icao and iata codes
4. 500: Internal server error

**Test Endpoint ->**

1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airline?iata=1t&icao=rnx**

Expected output ->

[

{

"name": "1Time Airline"

}

]

1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines**

Expected output ->

{

"error": "Please provide atleast one of ICAO or IATA"

}

1. **http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airline?iata=adasdas**

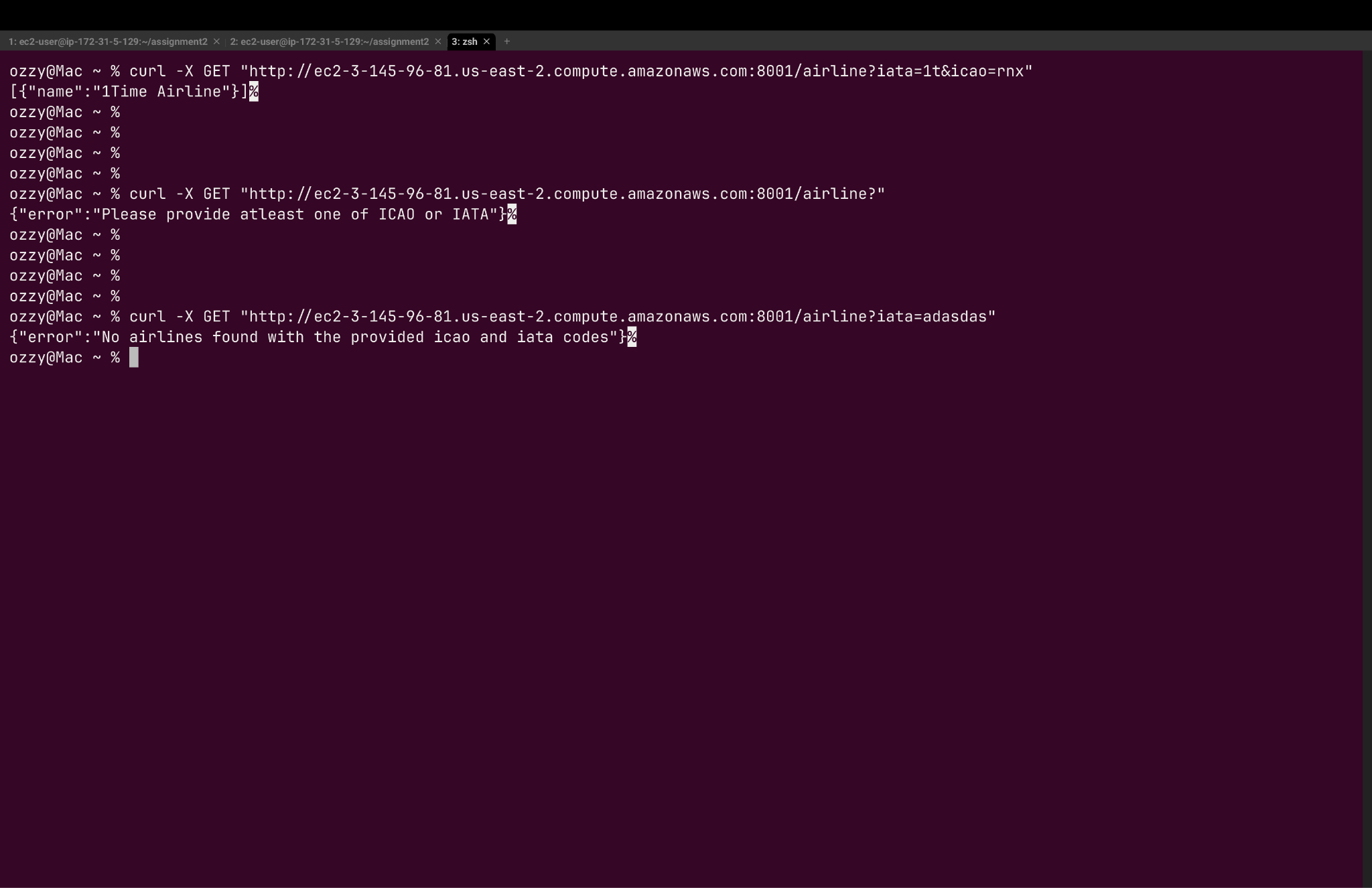
Expected output ->

{

"error": "No airlines found with the provided icao and iata codes"

}

Curl Image ->



**C) Add an airline**

* HTTP METHOD: **POST**
* URL -> **/airline**
* Description - Adds a new airline to the database.
* **Input Format**:

{

"name": "Test Airline",

"iata": "1G",

"icao": "AQE",

"callsign":

"TEST SIGN",

"country": "United States"

}

- **status Codes**:

1. 200: Airline created successfully!
2. 400: Name, Callsign and Country are required
3. 404: Airline with call sign -> ${callsign} already present!
4. 500: Internal server error

Test endpoint curl ->   
1) **curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines \**

**-H "Content-Type: application/json" \**

**-d '{**

**"name": "Test Airline2",**

**"iata": "2G",**

**"icao": "AQZ",**

**"callsign": "TEST SIGN3",**

**"country": "United States"**

**}'**

**Expected output** ->

{"status":"Airline created successfully!"}

2) **curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines \**

**-H "Content-Type: application/json" \**

**-d '{**

**"name": "Test Airline2",**

**"iata": "2G",**

**"icao": "AQZ",**

**"callsign": "TEST SIGN3",**

**"country": "United States"**

**}'**

// run the same curl command again

**Expected output** ->

{"error":"Airline with call sign -> TEST SIGN1 already present!"}

3) **curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines \**

**-H "Content-Type: application/json" \**

**-d '{**

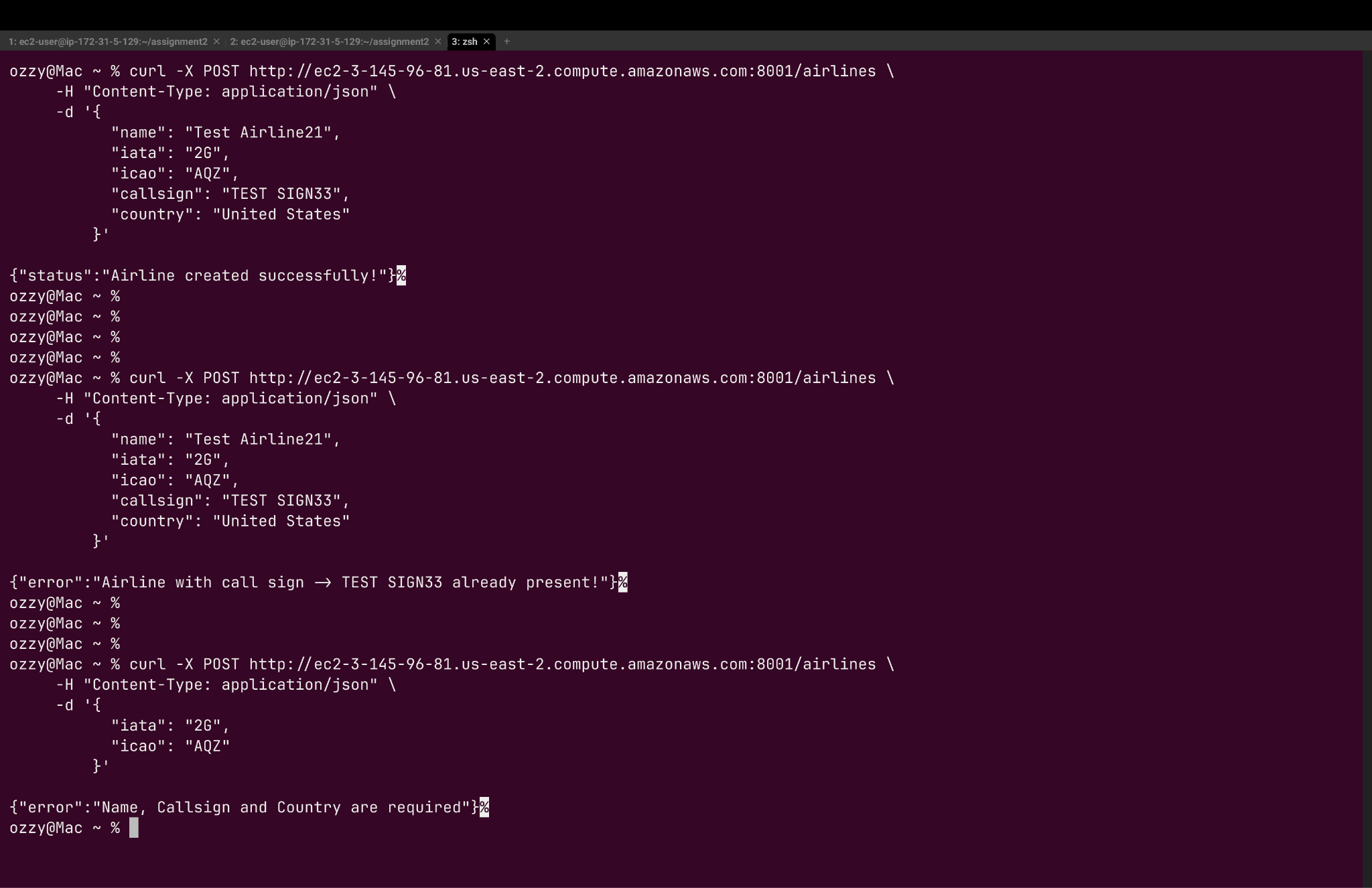
**"iata": "2G",**

**"icao": "AQZ"**

**}'**

**Expected output ->**{"error":"Name, Callsign and Country are required"}

Curl image ->



**D) Delete an Airline**

* HTTP METHOD: **DELETE**
* URL -> **iata=<iata code>&icao=<icao code>**
* Description - deletes an airline from the database.
* Input Parameters: A 3-letter ICAO code and or 2-letter IATA code
* **Output Format:**

{ status: Airline deleted successfully! }

- **status Codes**:

1. 200: Airline deleted successfully!
2. 400: need at least icao or iata to delete
3. 404: No airline were deleted with the provided icao and/or iata codes
4. 500: Internal server error

Test endpoint curl ->

1. **curl -X DELETE "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?iata=2G&icao=AQZ"**

**Expected output** ->

{"status":"Airline deleted successfully!"}%

1. **curl -X DELETE "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?"**

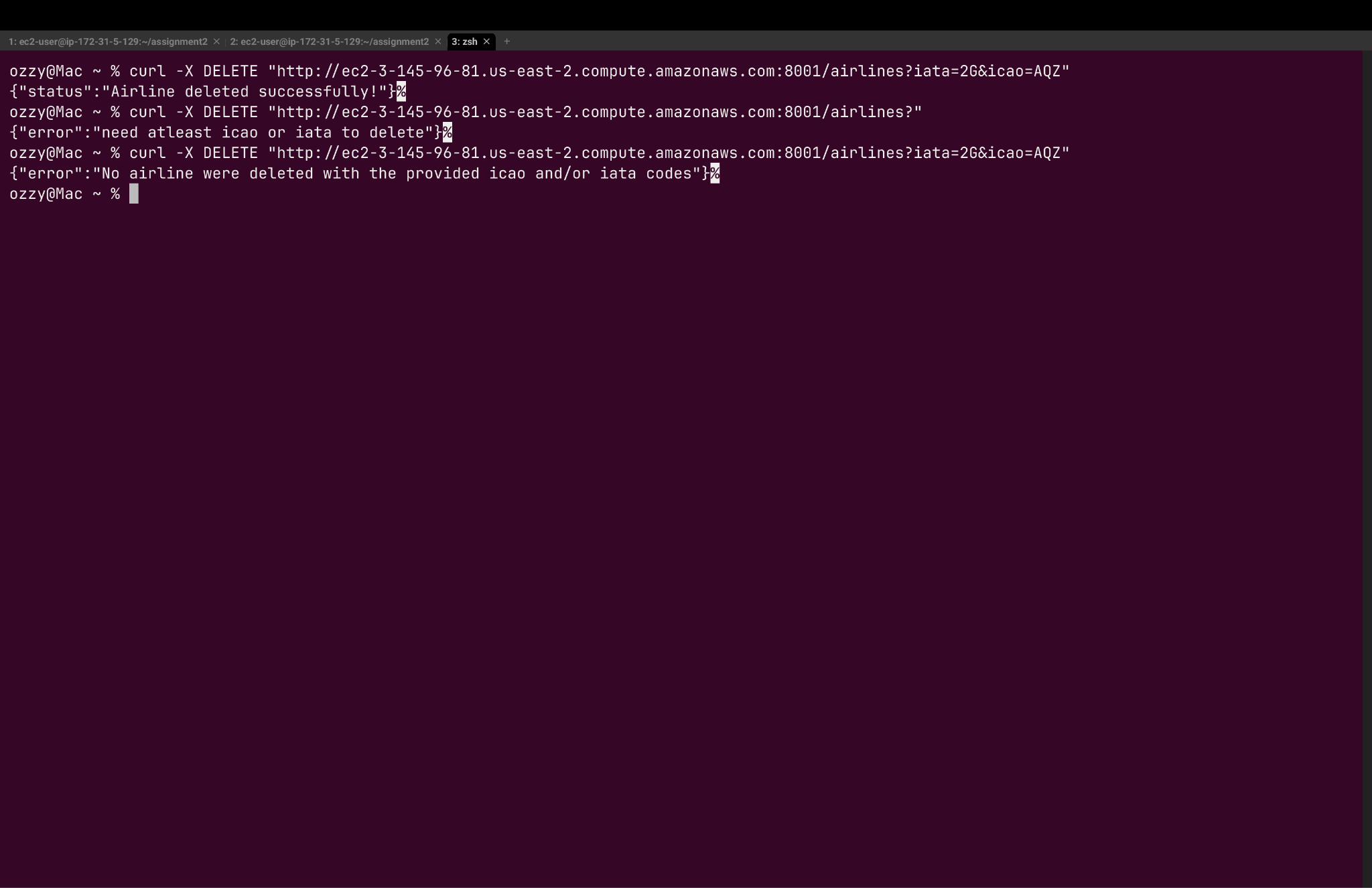
**Expected output** ->

{"error":"need atleast icao or iata to delete"}%

1. **curl -X DELETE "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?iata=2G&icao=AQZ**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airlines?iata=2G&icao=AQZ)**"**

**Expected output** ->

{"error":"No airline were deleted with the provided icao and/or iata codes"}%

Curl Image -> 

**2) Airports CRUD endpoints**

1. **Get airports by country codes**

* Method: **GET**
* URL: /airports?country=COUNTRY\_CODE1,COUNTRY\_CODE2,...
* Description: Returns a list of airports located in the specified countries.
* **Parameters:** country (required) comma-separated list of country codes
* **Output Format:**

[

{

"name": "V.C. Bird International Airport",

"city": "Antigua",

"country": "Antigua and Barbuda",

"iata": "ANU",

"icao": "TAPA",

"latitude": 17.1367,

"longitude": -61.7927

},

]

* **status Codes:**

1. 200: List of airports from the specified countries
2. 400: Missing country parameter
3. 404: No airports found for the specified country codes
4. 500: Internal server error

Test endpoint curl ->

1. curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports?country=bq,aw,ag"

**Expected output** ->

[{"name":"V.C. Bird International Airport","city":"Antigua","country":"Antigua and Barbuda","iata":"ANU","icao":"TAPA","latitude":17.1367,"longitude":-61.7927},{"name":"Queen Beatrix International Airport","city":"Oranjestad","country":"Aruba","iata":"AUA","icao":"TNCA","latitude":12.5014,"longitude":-70.0152},{"name":"Codrington Airport","city":"Codrington","country":"Antigua and Barbuda","iata":"BBQ","icao":"TAPH","latitude":17.6358,"longitude":-61.8286}]%

1. curl -X GET "<http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports?country>"

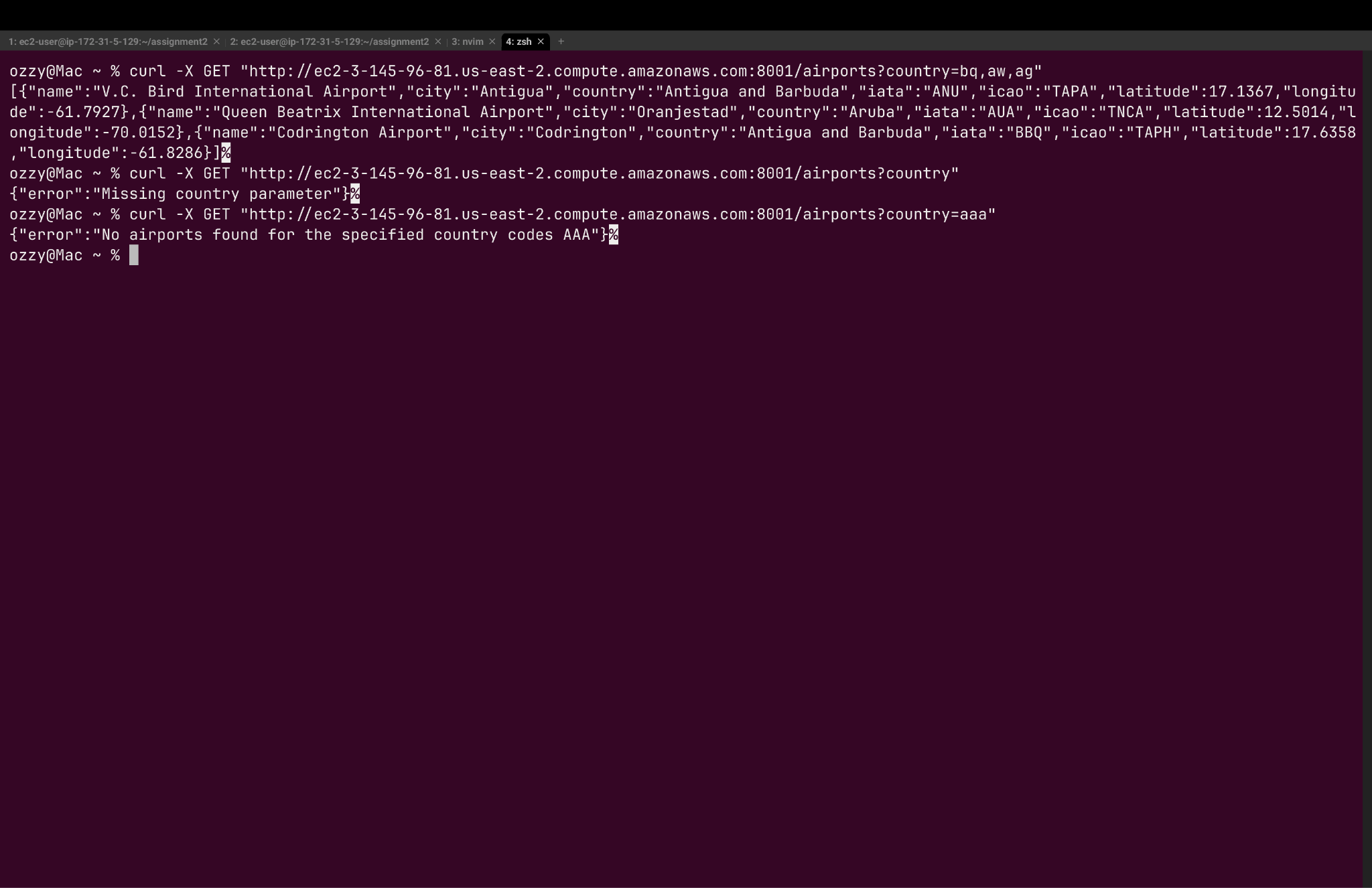
**Expected output** ->

{"error":"Missing country parameter"}%

1. curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports?country=aaa"

**Expected output** ->

{"error":"No airports found for the specified country codes AAA"}%

Curl Image -> 

1. **Get airport by ICAO and/or IATA code (with weather information)**

* Method: **GET**
* URL: /airport?icao=ICAO\_CODE&iata=IATA\_CODE
* Description: Returns a specific airport based on its ICAO and/or IATA code, including current weather information
* **Parameters:** icao (optional) the ICAO code of the airport and/or iata (optional) the IATA code of the airport. At least one of icao or iata must be provided for the endpoint to work
* **Output Format:**

{

"name": "La Guardia Airport",

"city": "New York",

"country": "United States",

"iata": "LGA",

"icao": "KLGA",

"latitude": 40.7772,

"longitude": -73.8726,

"high": 26.8,

"low": 6.5

}

* **status Codes:**

1. 200: Information about the specified airport with weather data
2. 400: Missing both ICAO and IATA parameters
3. 404: No airport found with the provided codes
4. 500: Internal server error

Test endpoint curl ->

1. curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airport?iata=lga&icao=klga"

**Expected output** ->

{"name":"La Guardia Airport","city":"New York","country":"United States","iata":"LGA","icao":"KLGA","latitude":40.7772,"longitude":-73.8726,"high":26.8,"low":6.5}%

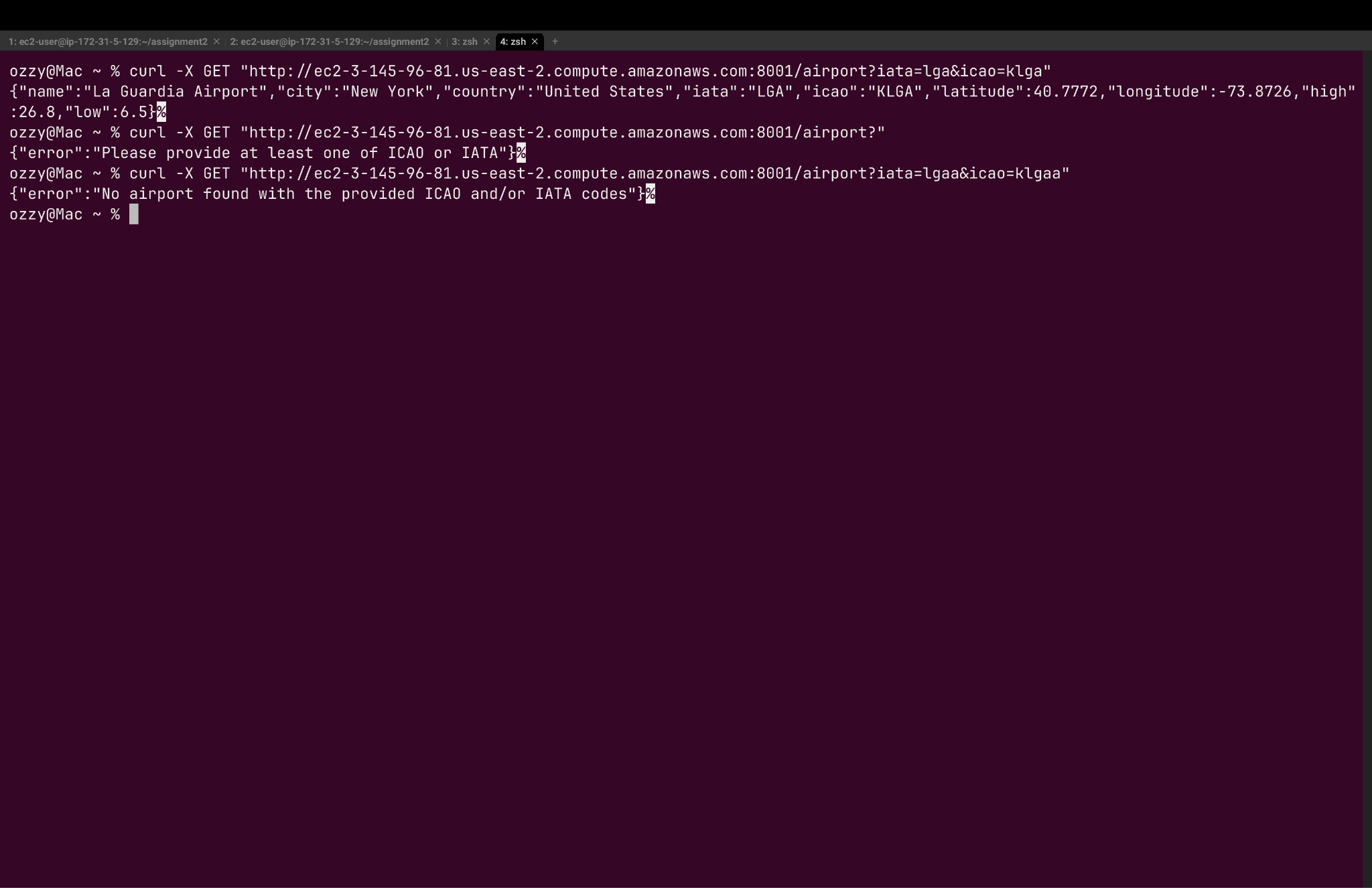
1. curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airport?"

**Expected output** ->

{"error":"Please provide at least one of ICAO or IATA"}%

1. curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airport?iata=lgaa&icao=klgaa"

**Expected output** -> {"error":"No airport found with the provided ICAO and/or IATA codes"}%

Curl Image -> 

1. **Add a new airport**

* Method: **POST**
* URL: /airports
* Description: Adds a new airport to the database
* **Input Format:**

{

"name": "Test Airport",

"city": "Test City",

"country": "Test Country",

"iata": "XXE",

"icao": "XXEE",

"latitude": 41.123456,

"longitude": -71.123456

}'

* **status Codes:**

1. 200: Airport created successfully
2. 400: Missing required fields
3. 404: Airport with IATA/ICAO code already exists
4. 500: Internal server error

Test endpoint curl ->

1. curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports \

-H "Content-Type: application/json" \

-d '{

"name": "Test Airport",

"city": "Test City",

"country": "Test Country",

"iata": "XXE",

"icao": "XXEE",

"latitude": 41.123456,

"longitude": -71.123456

}'

**Expected output** ->

{"status":"Airport created successfully!"}%

1. curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports \

-H "Content-Type: application/json" \

-d '{

"name": "Test Airport",

"city": "Test City",

"country": "Test Country",

"iata": "XXE",

"icao": "XXEE",

"latitude": 41.123456,

"longitude": -71.123456

}'

**Expected output** ->

{"error":"Airport with IATA code -> XXE already exists!"}%

1. curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports \

-H "Content-Type: application/json" \

-d '{

"iata": "XXE",

"icao": "XXEE",

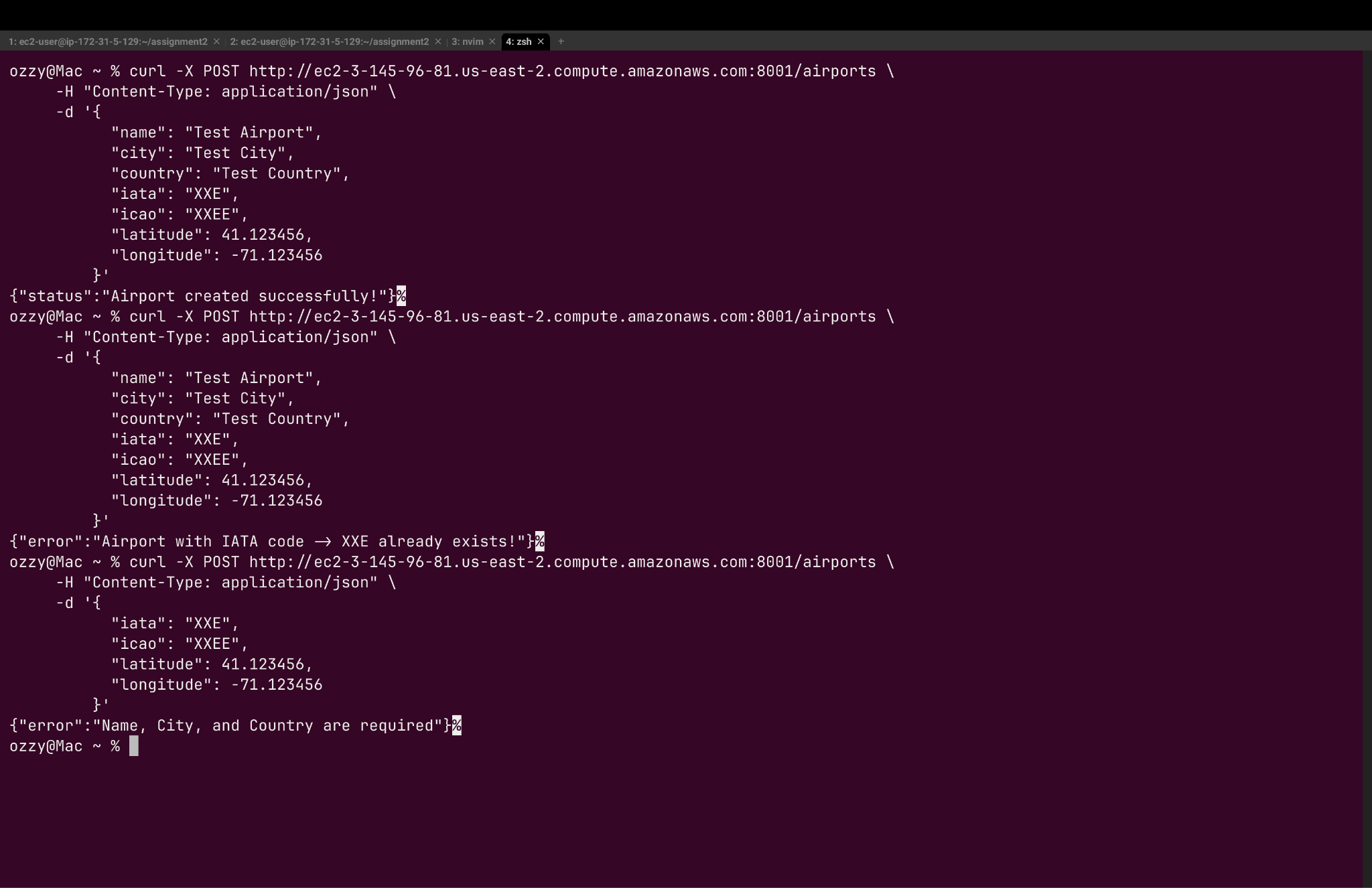
"latitude": 41.123456,

"longitude": -71.123456

}'

**Expected output** ->

{"error":"Name, City, and Country are required"}%

Curl Image -> 

1. **Delete an airport**

* Method: **DELETE**
* URL: /airports?icao=ICAO\_CODE&iata=IATA\_CODE
* Description: Deletes an airport based on its ICAO and/or IATA code.
* **Parameters:** icao (optional) the ICAO code of the airport and/or iata (optional) the IATA code of the airport, At least one of icao or iata must be provided
* **Output Format:**

{

"status":"Airport deleted successfully!"

}

* **status Codes:**

1. 200: Airport deleted successfully
2. 400: Missing both ICAO and IATA parameters
3. 404: no airport deleted
4. 500: Internal server error

Test endpoint curl ->

1. curl -X DELETE "<http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports?icao=XXEE&iata=XXE>"

**Expected output** ->

{"status":"Airport deleted successfully!"}%

1. curl -X DELETE "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports?icao=XXEE&iata=XXE"

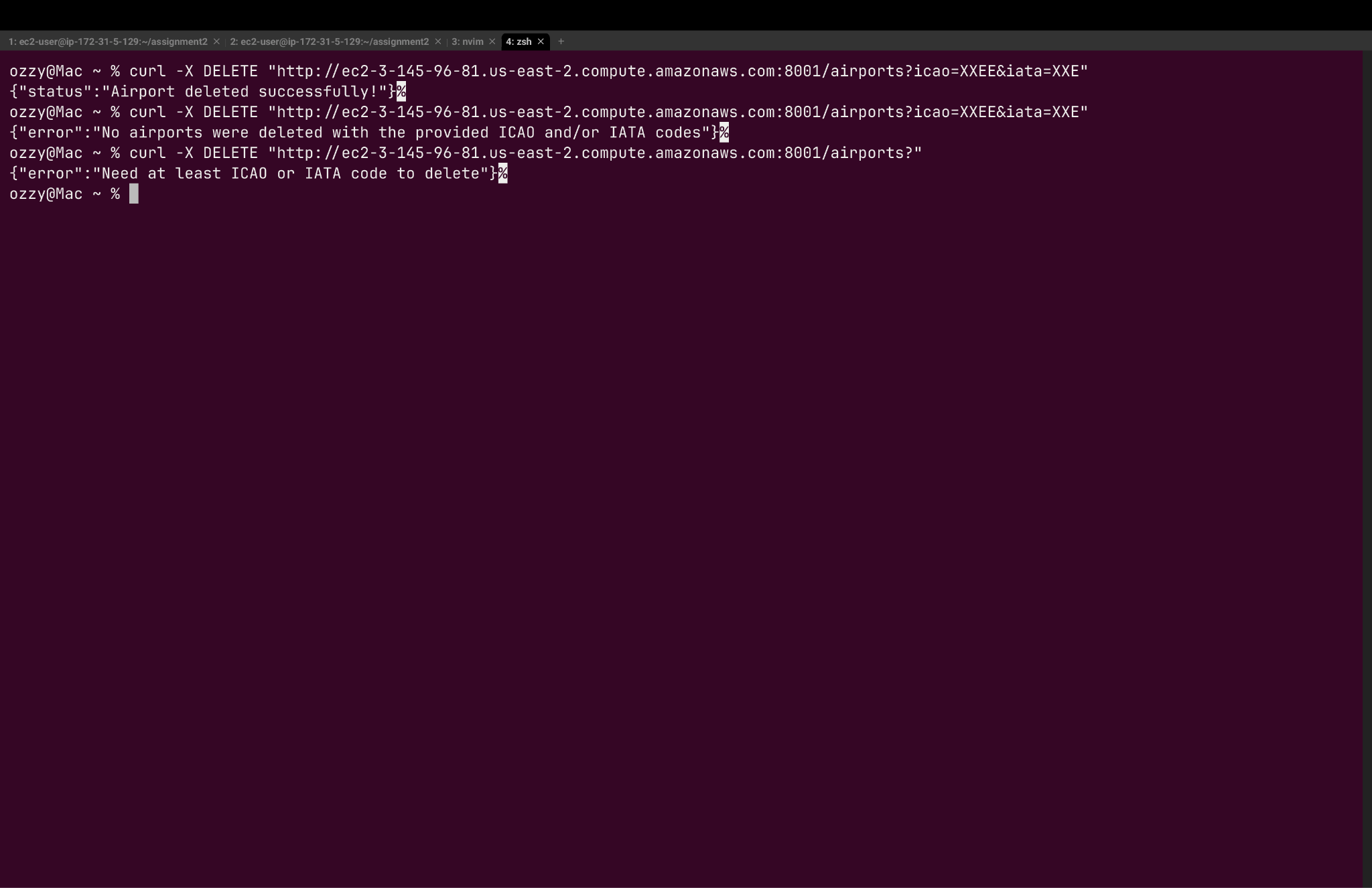
**Expected output** ->

{"error":"No airports were deleted with the provided ICAO and/or IATA codes"}%

1. curl -X DELETE "<http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/airports>?"

**Expected output** ->

{"error":"Need at least ICAO or IATA code to delete"}%

**Curl Image -> **

**3) Routes CRUD endpoint**

1. **Calculate distance and get airlines/aircraft**

* Method: **GET**
* URL: /routes?departure=DEP\_IATA&arrival=ARR\_IATA
* Description: Calculates the distance between two airports using the Haversine formula and returns the list of airlines and aircraft types for that route.
* **Parameters:** departure (required): IATA code (3-character) of the departure airport and arrival (required): IATA code (3-character) of the arrival airport
* **Output Format:**

{

"departure": {

"code": "JFK",

"name": "John F Kennedy International Airport"

},

"arrival": {

"code": "LAX",

"name": "Los Angeles International Airport"

},

"distance": 3978.56,

"unit": "km",

"routes": [

{

"airline": "AA ",

"aircraft\_types": [

"32B",

"762"

]

}

]

}

* **status Codes:**

1. 200: Distance between the airports and available routes
2. 400: Both departure and arrival airport codes are required
3. 404: One or both airports not found
4. 500: Internal server error

Test endpoint curl ->

1. **curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?departure=JFK&arrival=LAX"**

**Expected output** ->

{"departure":{"code":"JFK","name":"John F Kennedy International Airport"},"arrival":{"code":"LAX","name":"Los Angeles International Airport"},"distance":3978.56,"unit":"km","routes":[{"airline":"AA ","aircraft\_types":["32B","762"]},{"airline":"AS ","aircraft\_types":["321","762"]},{"airline":"B6 ","aircraft\_types":["320"]},{"airline":"DL ","aircraft\_types":["76W","752"]},{"airline":"QF ","aircraft\_types":["744"]},{"airline":"UA ","aircraft\_types":["757"]},{"airline":"US ","aircraft\_types":["32B","762"]},{"airline":"VX ","aircraft\_types":["320"]}]}%

1. **curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?"**

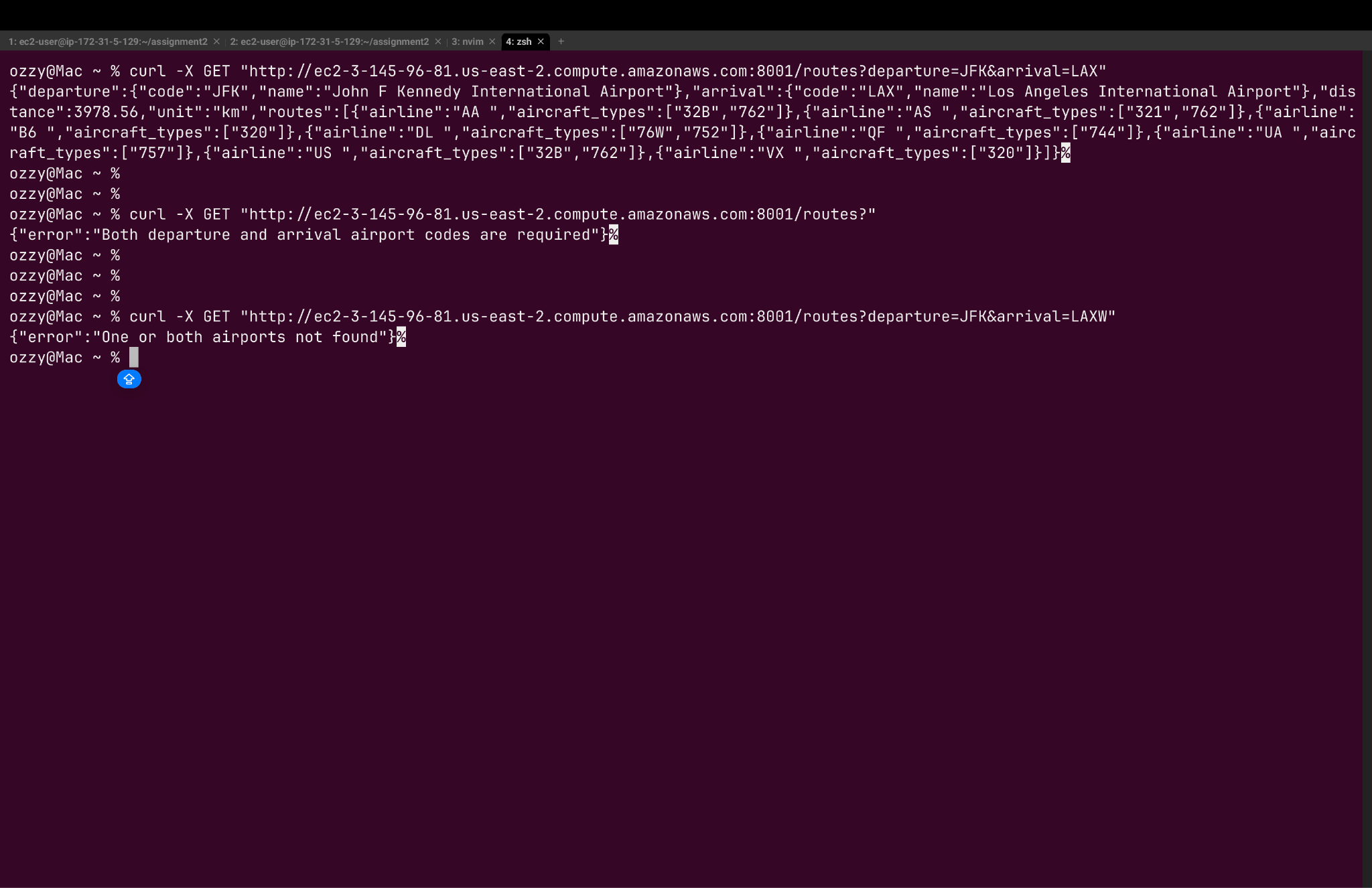
**Expected output** ->

{"error":"Both departure and arrival airport codes are required"}%

1. **curl -X GET "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?departure=JFK&arrival=LAXW**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?departure=JFK&arrival=LAXW)**"**

**Expected output** ->

{"error":"One or both airports not found"}%

Curl Image -> 

**B) Get arrival airports from departure**

* Method: **GET**
* URL: /routes/arrivals?departure=DEP\_IATA
* Description: Returns a list of all arrival airports that can be reached from a specified departure airport.
* **Parameters:** departure (required): IATA code (3-character) of the departure airport
* **Output Format:**

[

{

"name": "Norman Y. Mineta San Jose International Airport",

"city": "San Jose",

"country": "United States",

"iata": "SJC",

"icao": "KSJC"

}

]

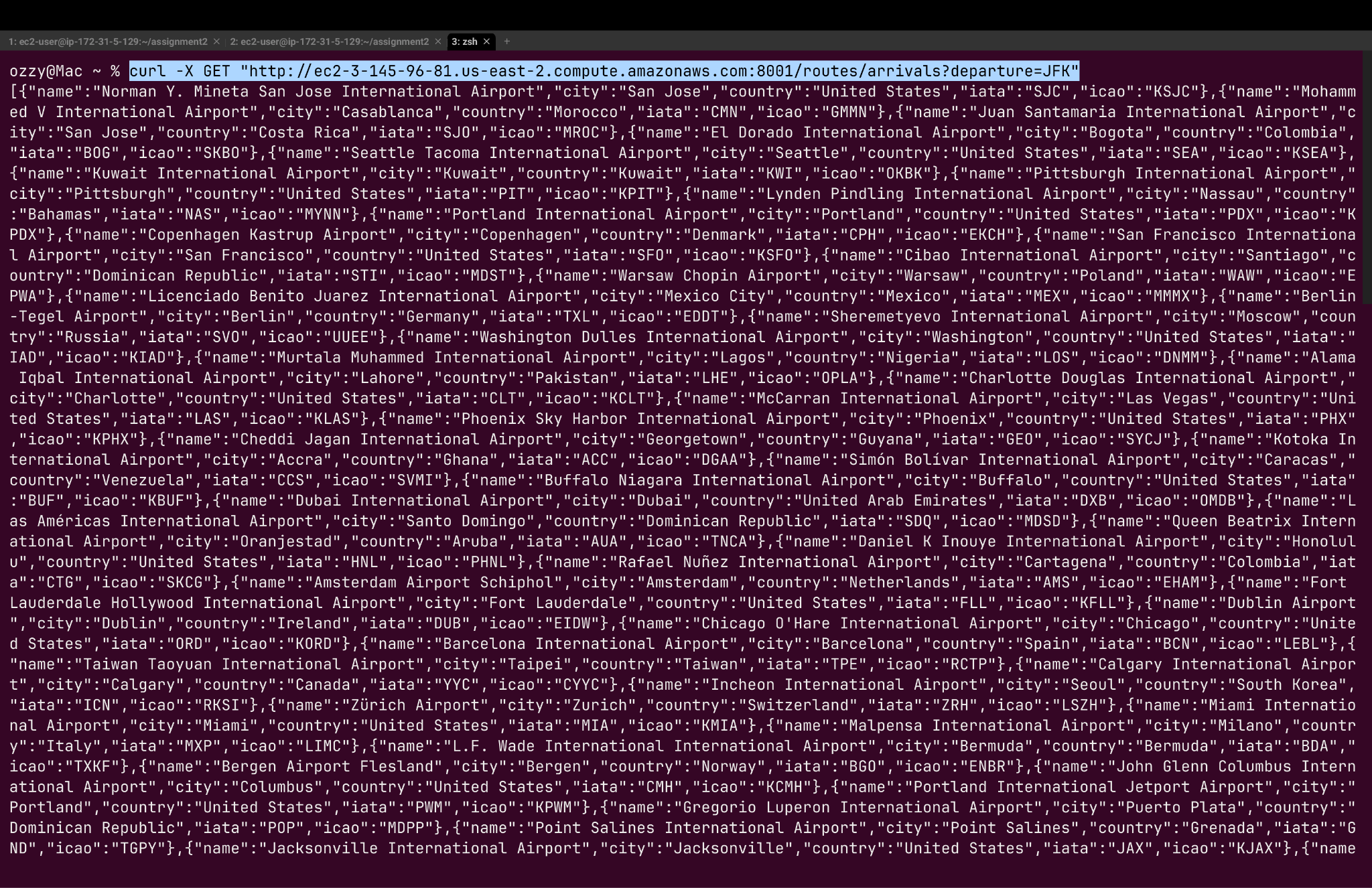
* **status Codes:**

1. 200: List of arrival airports
2. 400: Departure airport code is required
3. 404: No routes found from airport with code ${departure}
4. 500: Internal server error

Test endpoint curl ->

1. **curl -X GET "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/arrivals?departure=JFK**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/arrivals?departure=JFK)**"**

Expected output image (since output is too large)->



1. **curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/arrivals?"**

**Expected output** ->

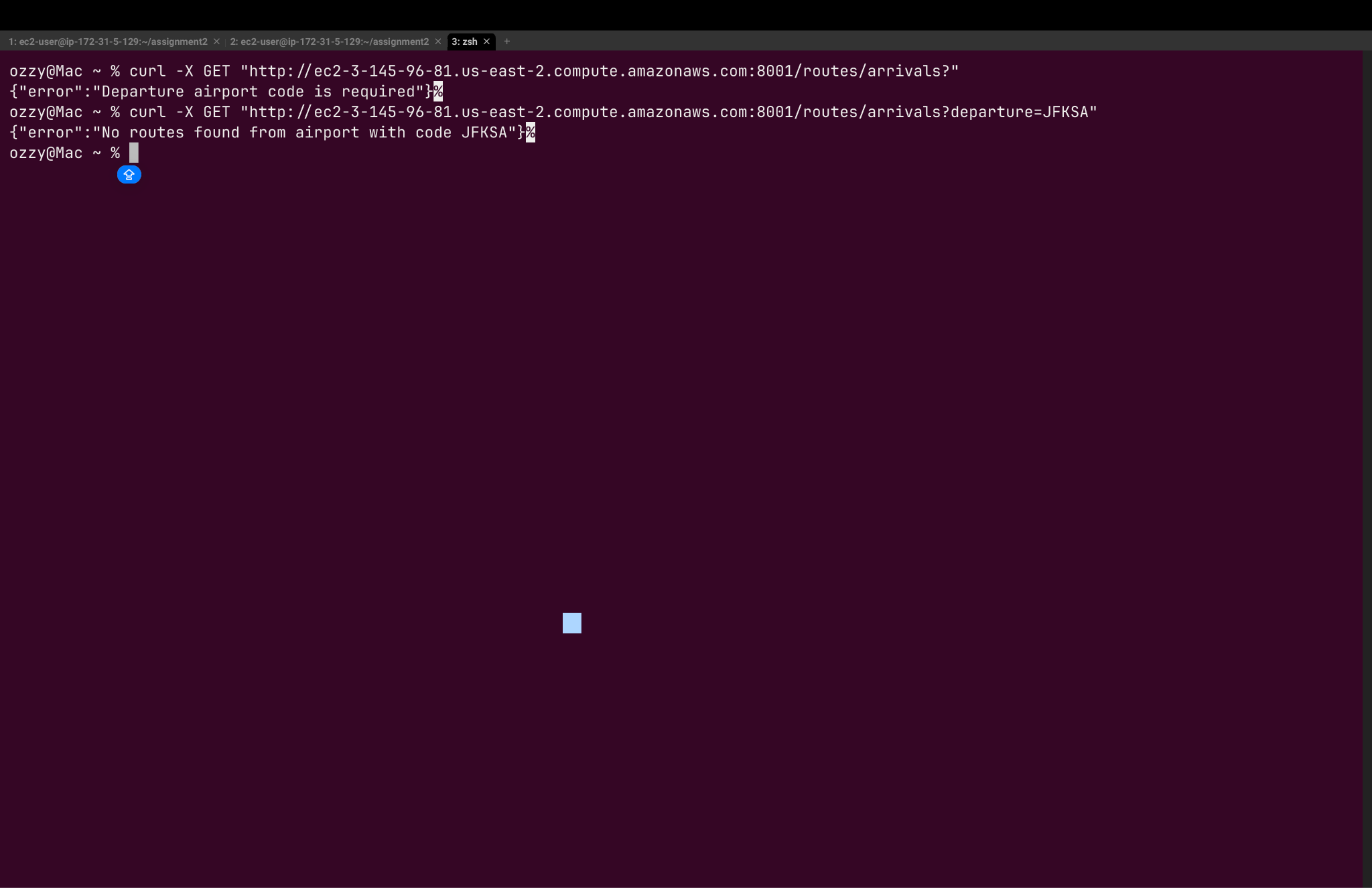
{"error":"Departure airport code is required"}%

1. **curl -X GET "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/arrivals?departure=JFKSA**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/arrivals?departure=JFKSA)**"**

**Expected output** ->

{"error":"No routes found from airport with code JFKSA"}%

Curl image of the remaining end points ->



**C) Get routes by airline and aircraft**

* Method: **GET**
* URL: /routes/byairline?airline=AIRLINE\_IATA&aircraft=AIRCRAFT\_CODE
* Description: Returns a list of routes (departure-arrival pairs) flown by a specified airline using a specific aircraft type.
* **Parameters:** airline (required): IATA code (2-character) of the airline and aircraft (required) Code (3-character) of the aircraft type
* **Output Format:**

[

{

"departure": "NBC",

"departure\_name": "Begishevo Airport",

"arrival": "GYD",

"arrival\_name": "Heydar Aliyev International Airport"

},

{

"departure": "NBC",

"departure\_name": "Begishevo Airport",

"arrival": "LED",

"arrival\_name": "Pulkovo Airport"

},

]

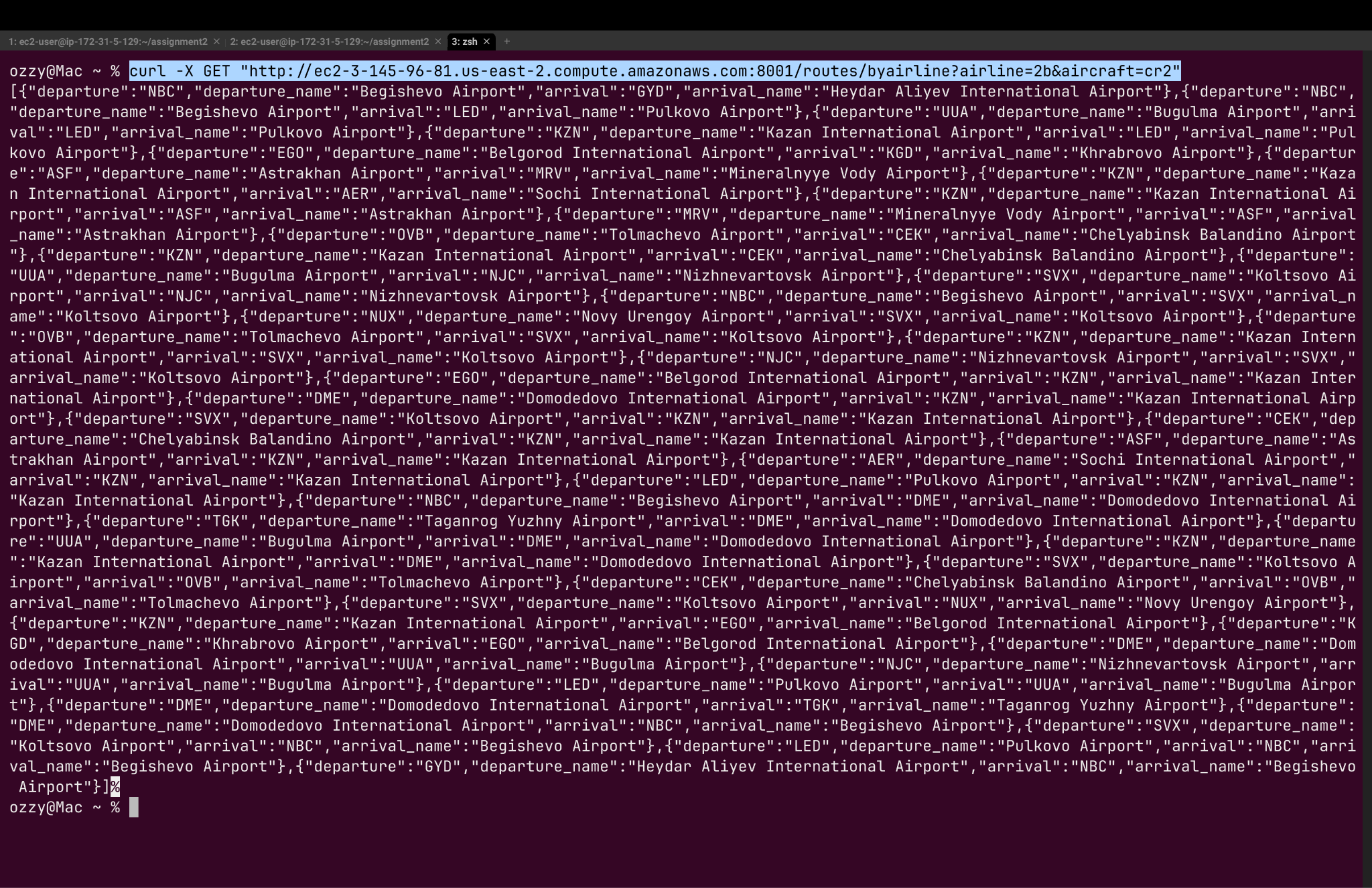
* **status Codes:**

1. 200: List of routes
2. 400: Both airline IATA code and aircraft type code are required
3. 404: No routes found for airline ${airline} using aircraft type ${aircraft}
4. 500: Internal server error

Test endpoint curl ->

1. **curl -X GET "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/byairline?airline=2b&aircraft=cr2"**

Expected output (Too long since screenshot) ->



1. **curl -X GET "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/byairline?airline=2b**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/byairline?airline=2b)**"**

**Expected output ->**

{"error":"Both airline IATA code and aircraft type code are required"}%

1. **curl -X GET "**[**http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/byairline?airline=2b&aircraft=crsd2**](http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes/byairline?airline=2b&aircraft=crsd2)"

**Expected output ->**

{"error":"No routes found for airline 2b using aircraft type crsd2"}%

Remaining Curl Images ->



**D) Add a new route**

* Method: **POST**
* URL: /routes/
* Description: Adds a new route with airline, departure airport, arrival airport, and aircraft type information.
* **Input Format:**

{

"airline": "AA", -> Required field

"departure": "JFK", -> Required field

"arrival": "LAX", -> Required field

"aircraft": ["738", "320", "777"] -> Required field

}

* **Output Format:**

{

"status":"Route created successfully"

}

* **status Codes:**

1. 200: Route created successfully
2. 400: Missing required fields or route already exists
3. 404: Airline, airport, or aircraft not found
4. 500: Internal server error

Test endpoint curl ->

1. curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{"airline":"TP","departure":"LIS","arrival":"ICN","aircraft":"339"}'

**Expected output ->**

{"status":"Route created successfully"}%

**2)** curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{

"airline": "AA",

"departure": "JFK",

"arrival": "LAX",

"aircraft": ["738", "320", "777"]

}'

**Expected output ->**

{"error":"This route already exists"}%

**3)** curl -X POST http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{

"airline": "AA",

"departure": "JFK",

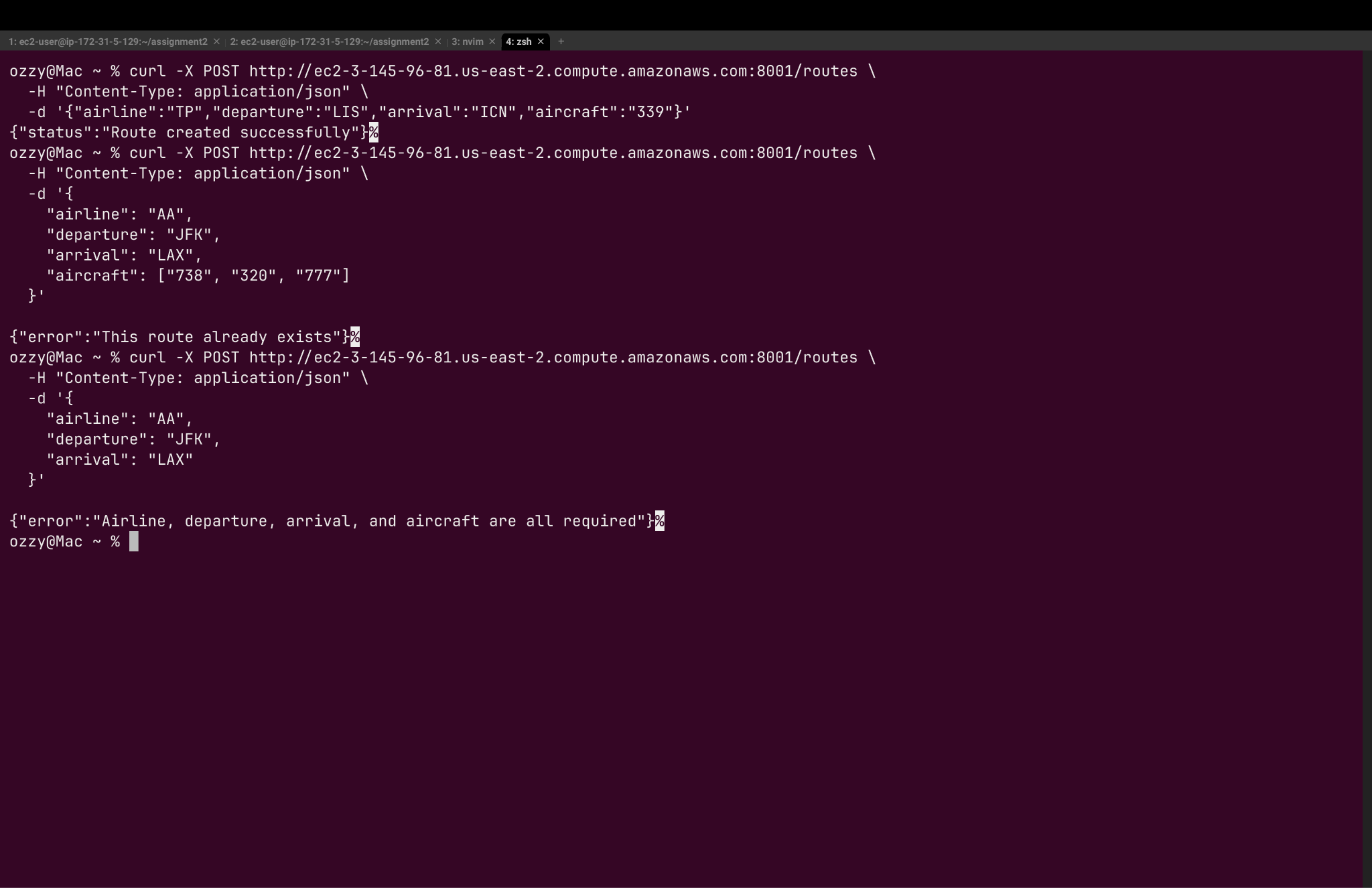
"arrival": "LAX"

}'

**Expected output ->**

{"error":"Airline, departure, arrival, and aircraft are all required"}%

Curl Image ->



**E) Delete a route**

* Method: **DELETE**
* URL: /routes?airline=AIRLINE\_IATA&departure=DEP\_IATA&arrival=ARR\_IATA
* Description: Returns a list of routes (departure-arrival pairs) flown by a specified airline using a specific aircraft type.
* **Parameters: airline** (required): IATA code (2-character) of the airline and **departure** (required): IATA code (3-character) of the departure airport and **arrival** (required): IATA code (3-character) of the arrival airport
* **Output Format:**
* **status Codes:**

1. 200: Route deleted successfully
2. 400: Missing parameters
3. 404: No route found matching the criteria
4. 500: Internal server error

Test endpoint curl ->

1. curl -X DELETE "<http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?airline=TP&departure=LIS&arrival=ICN>"

**Expected output ->**

{"status":"Route deleted successfully"}%

1. curl -X DELETE "http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?airline=TP&departure=LIS&arrival=ICN"

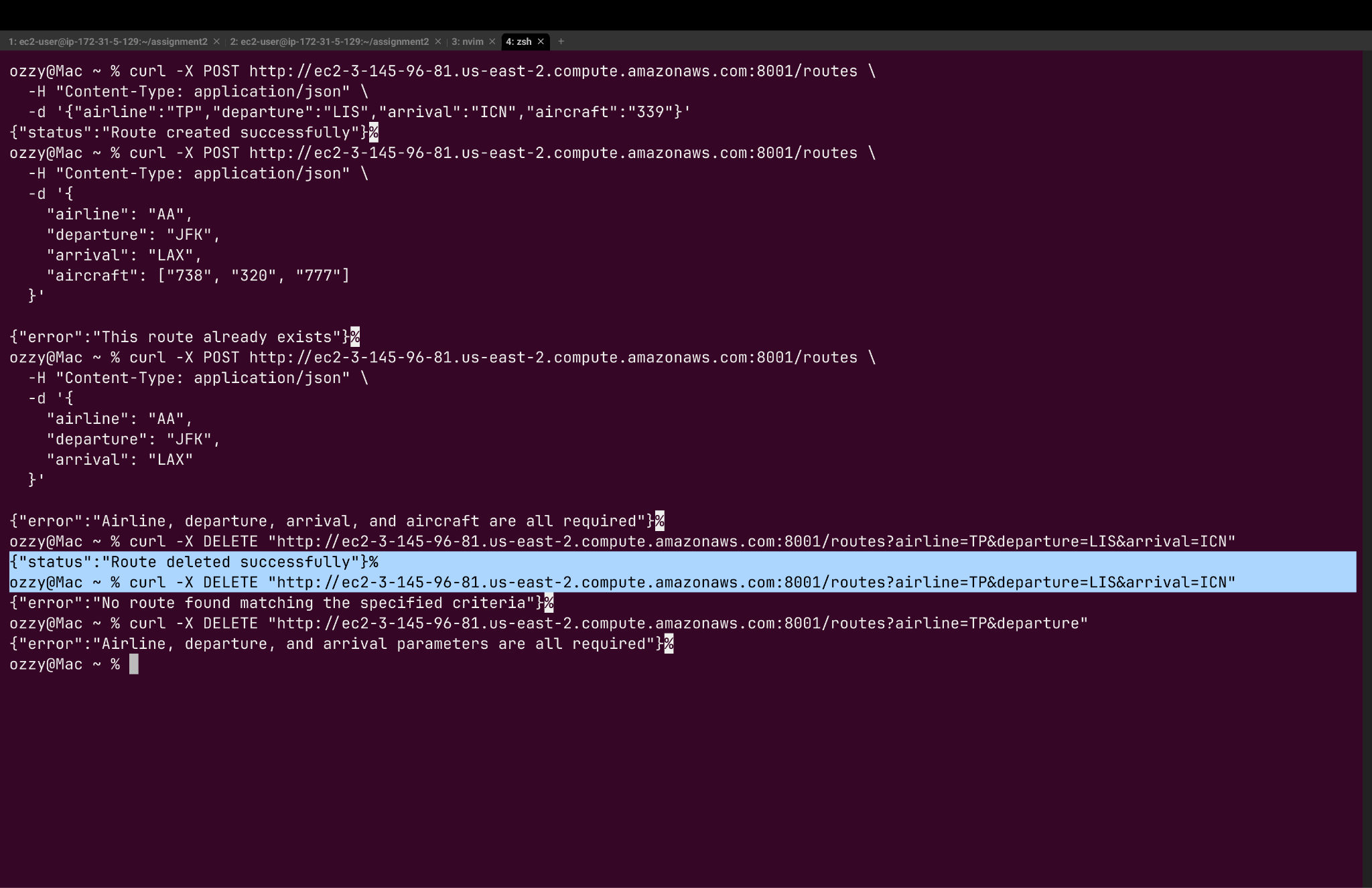
**Expected output ->**

{"error":"No route found matching the specified criteria"}%

1. curl -X DELETE "<http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes?airline=TP&departure>"

**Expected output ->**

{"error":"Airline, departure, and arrival parameters are all required"}%

Curl Image -> 

**E) Update a route with aircraft type**

* Method: **PUT**
* URL: /routes
* Description: Updates a route by adding a new aircraft type to the existing equipment list.
* **Input Format:**

{

"airline": "LH", -> Required Field

"departure": "FRA", -> Required Field

"arrival": "JFK", -> Required Field

"aircraft": "748" -> Required Field

}

* **Output Format:**

{

"status":"Route updated successfully",

"planes":"388 744 748"

}

* **status Codes:**

1. 200: Route updated successfully
2. 400: Missing required fields
3. 404: Route not found or aircraft type not found
4. 500: Internal server error

Test endpoint curl ->

1. curl -X PUT http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{

"airline": "LH",

"departure": "FRA",

"arrival": "JFK",

"aircraft": "748"

}'

**Expected output ->**

{"status":"Route updated successfully","planes":"388 744 748"}%

1. curl -X PUT http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{

"airline": "TP",

"departure": "LIS",

"arrival": "ICN",

"aircraft": "748"

}'

**Expected output ->**

{"error":"Route not found"}%

1. curl -X PUT http://ec2-3-145-96-81.us-east-2.compute.amazonaws.com:8001/routes \

-H "Content-Type: application/json" \

-d '{

"airline": "TP",

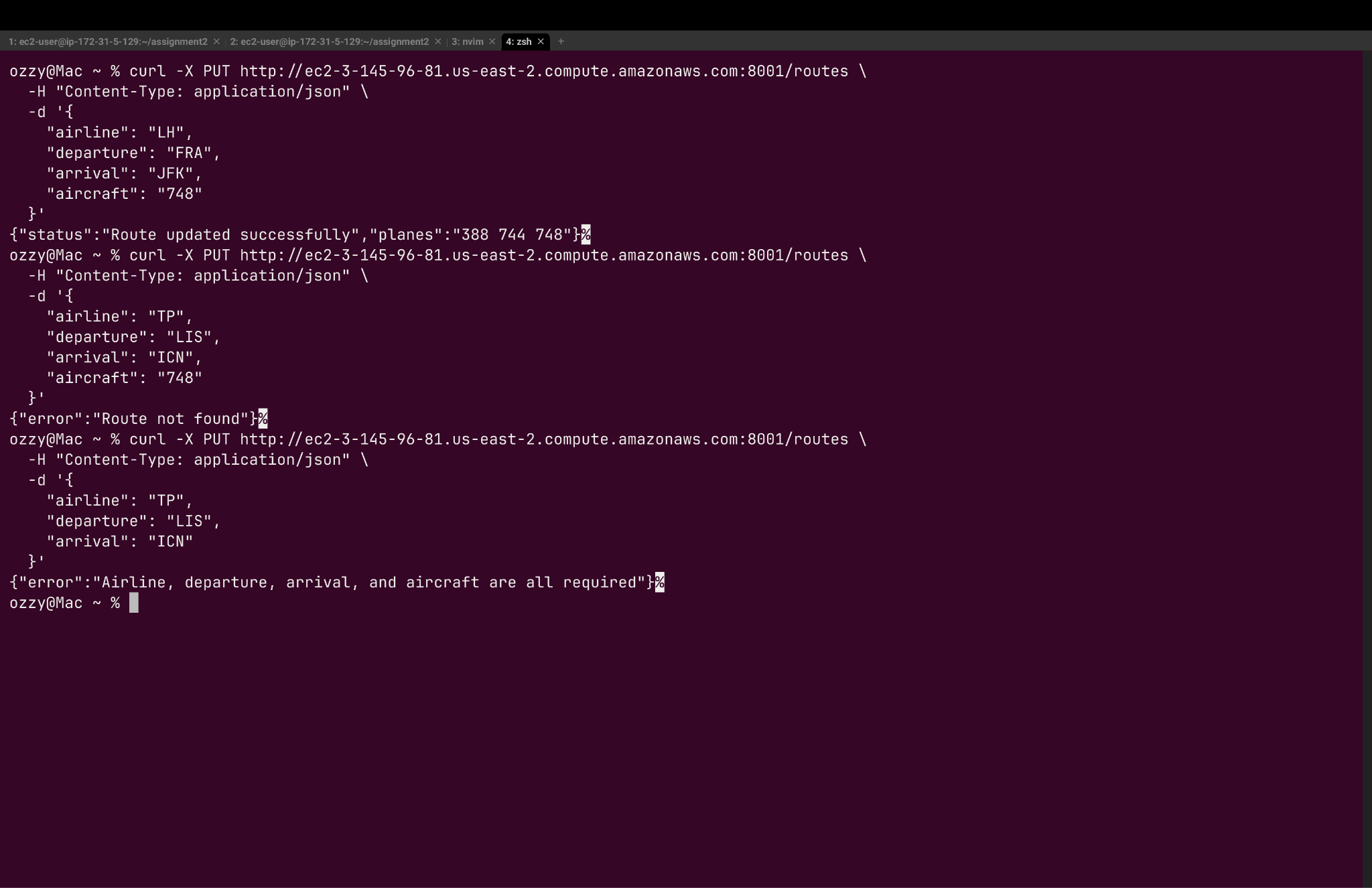
"departure": "LIS",

"arrival": "ICN"

}'

**Expected output ->**

{"error":"Airline, departure, arrival, and aircraft are all required"}%

**Curl Image -> **