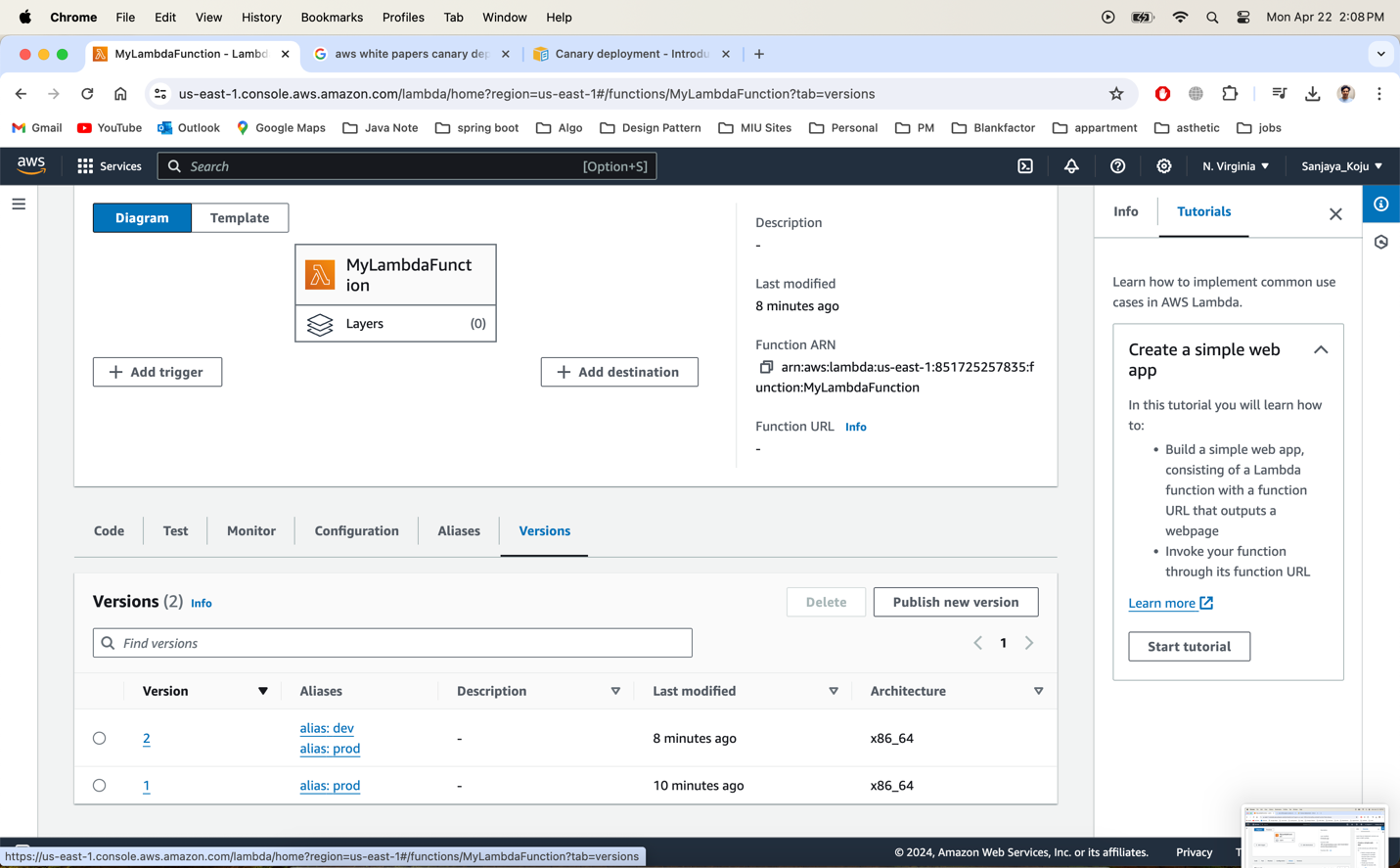
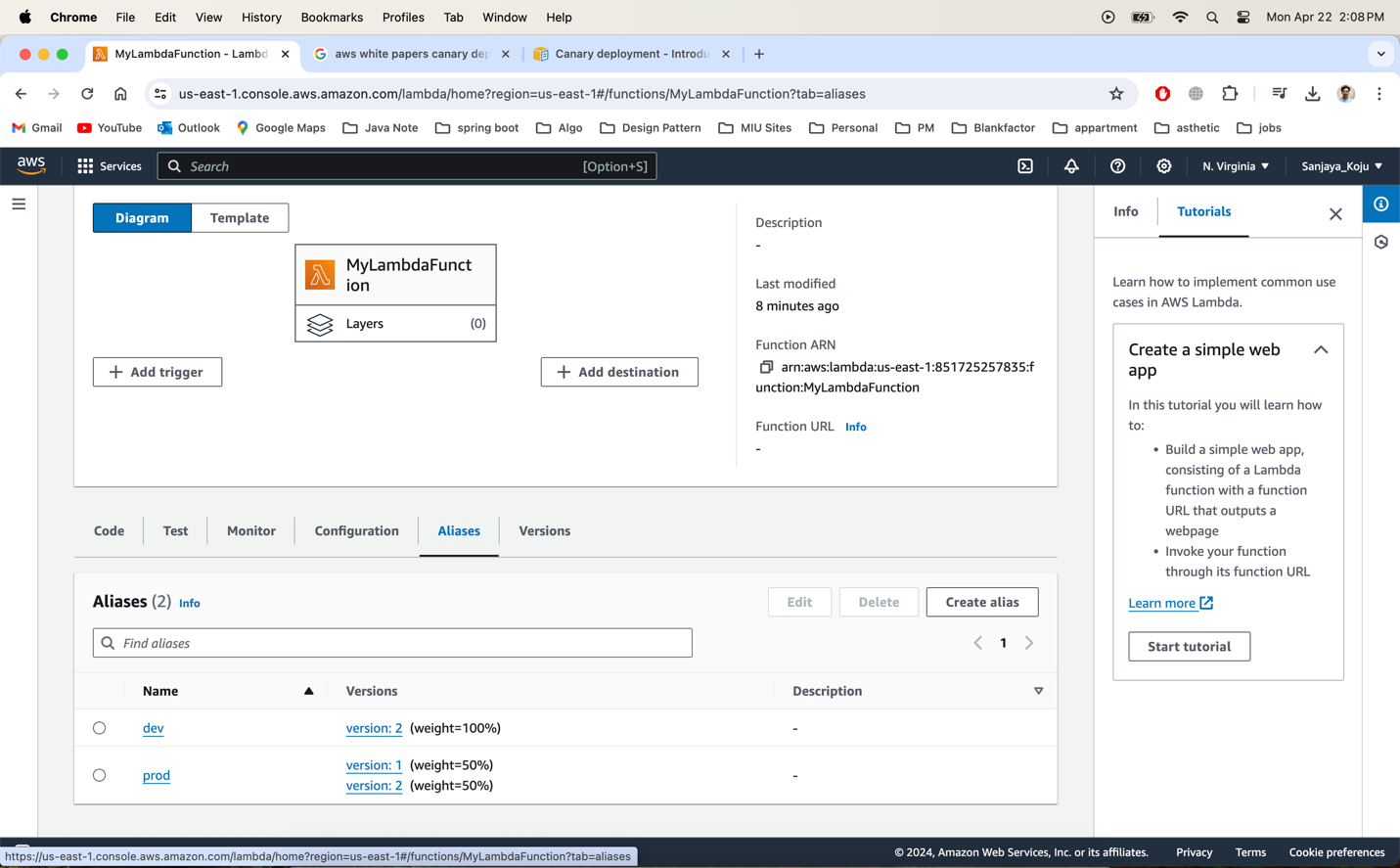
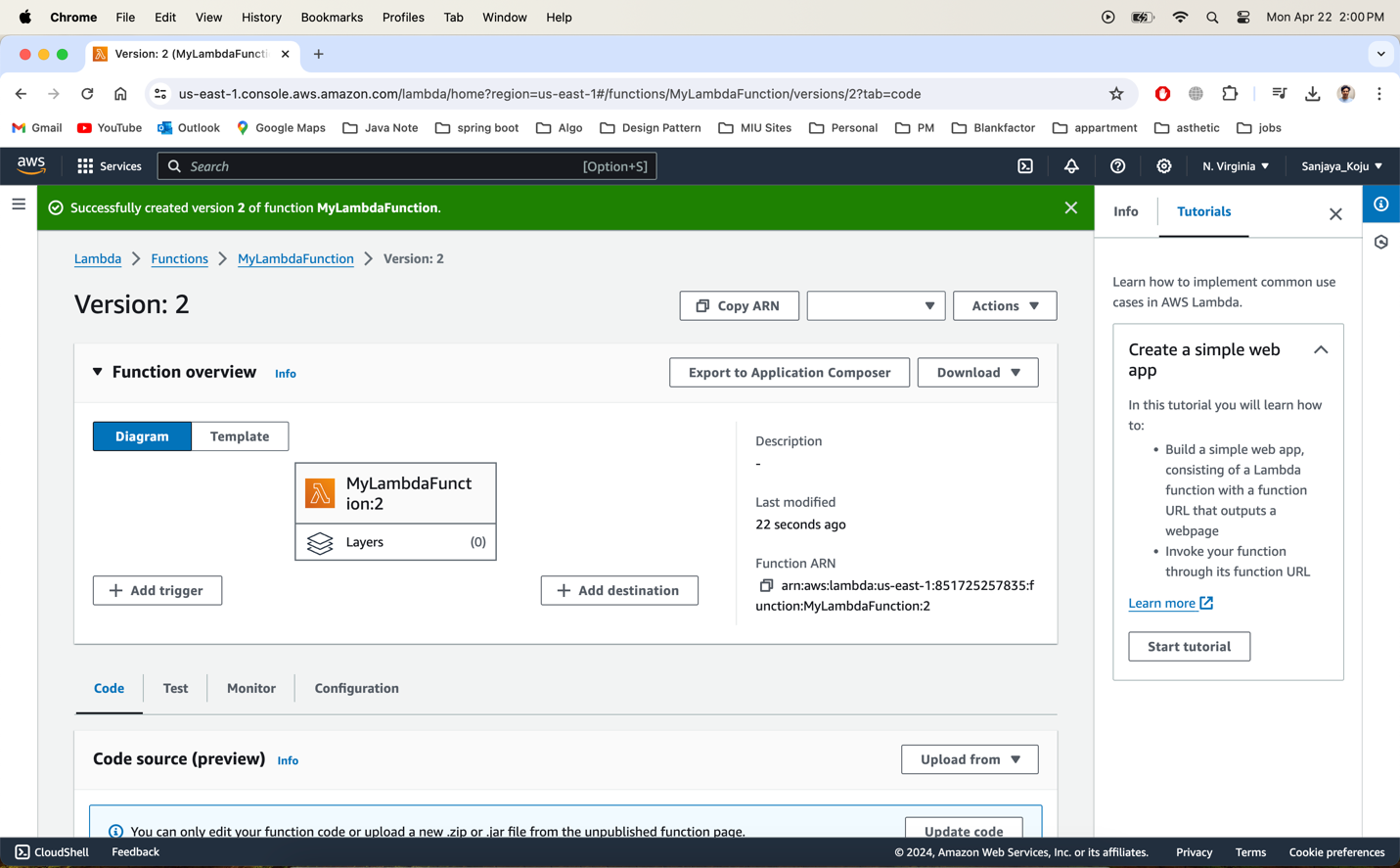
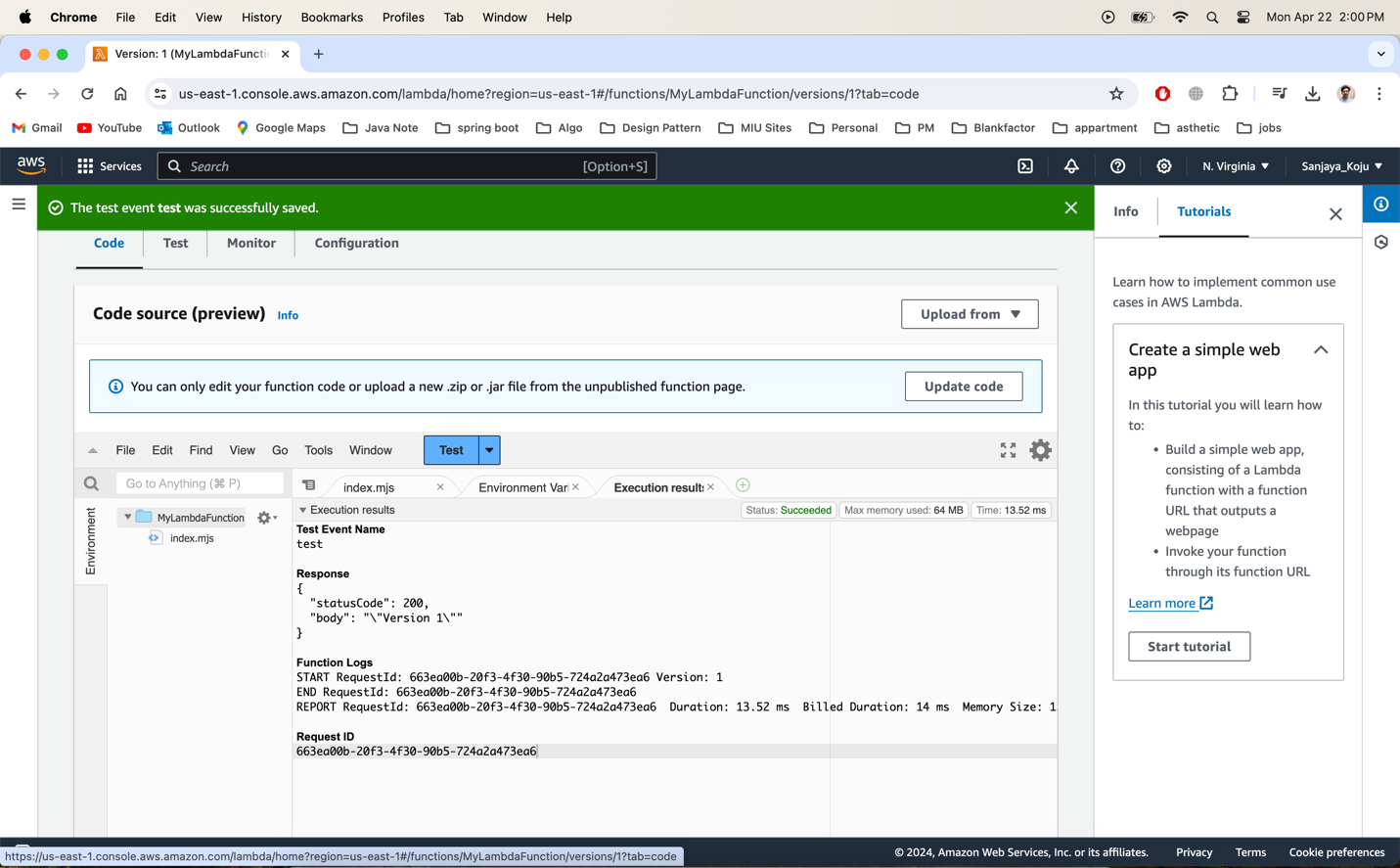
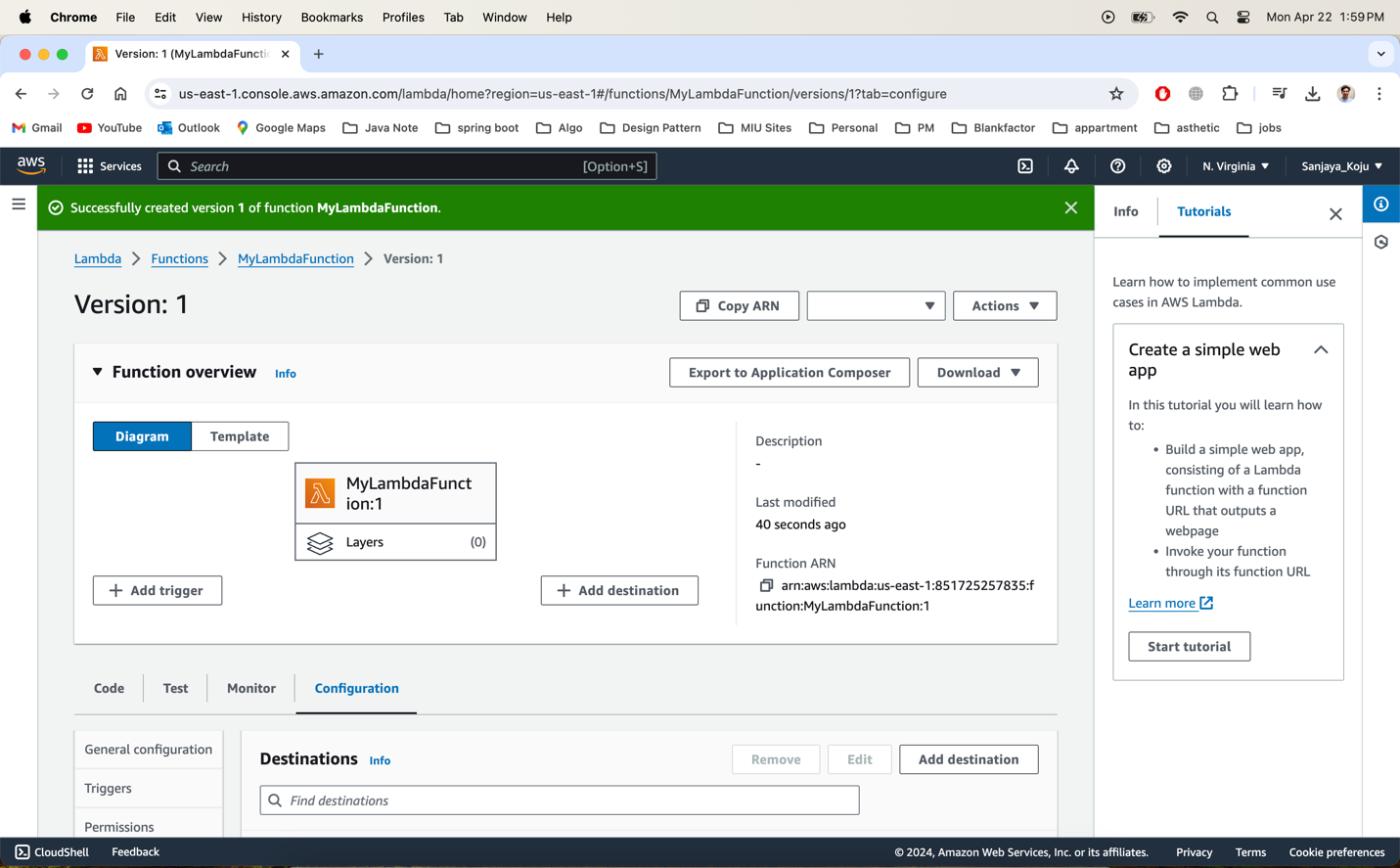
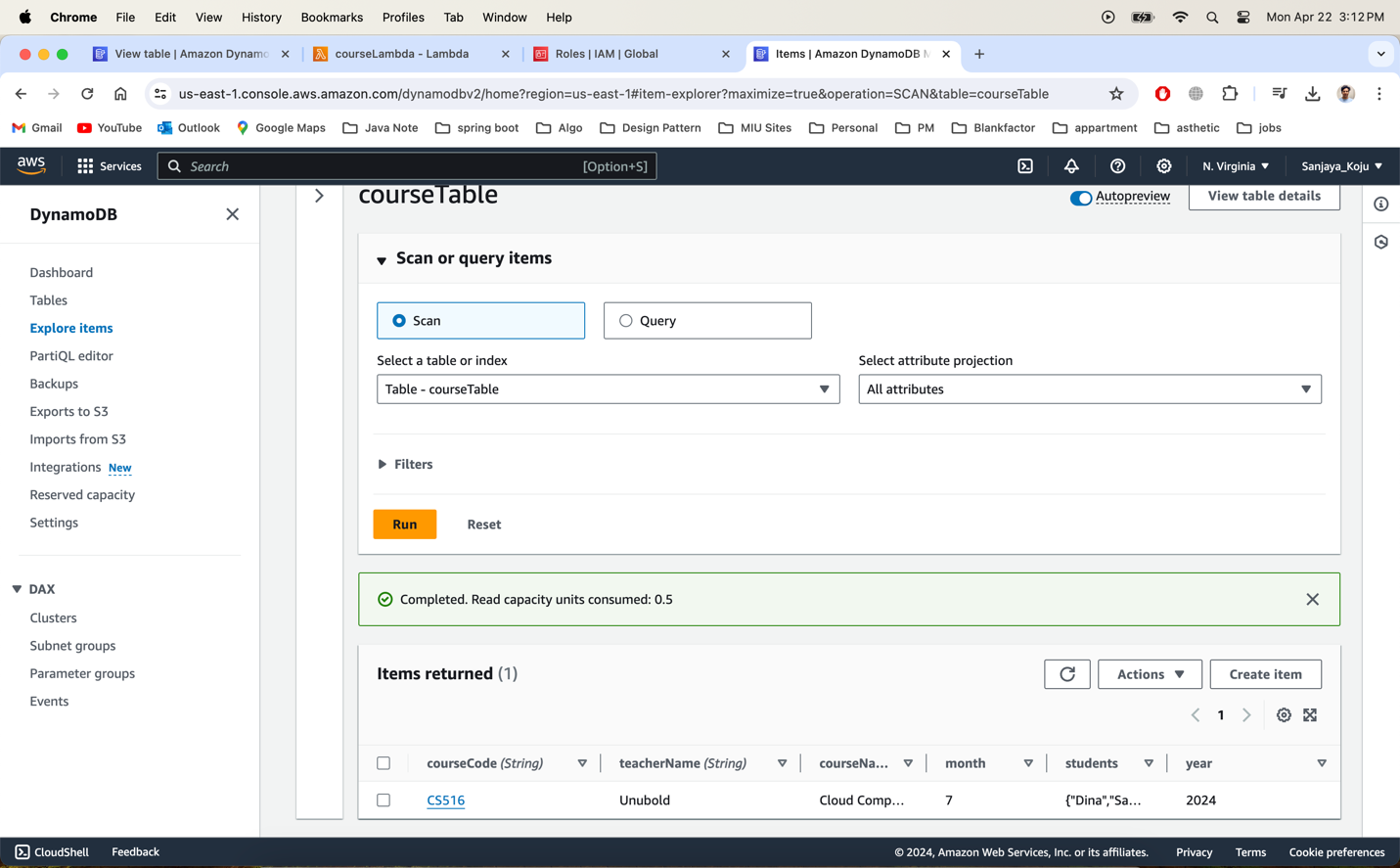
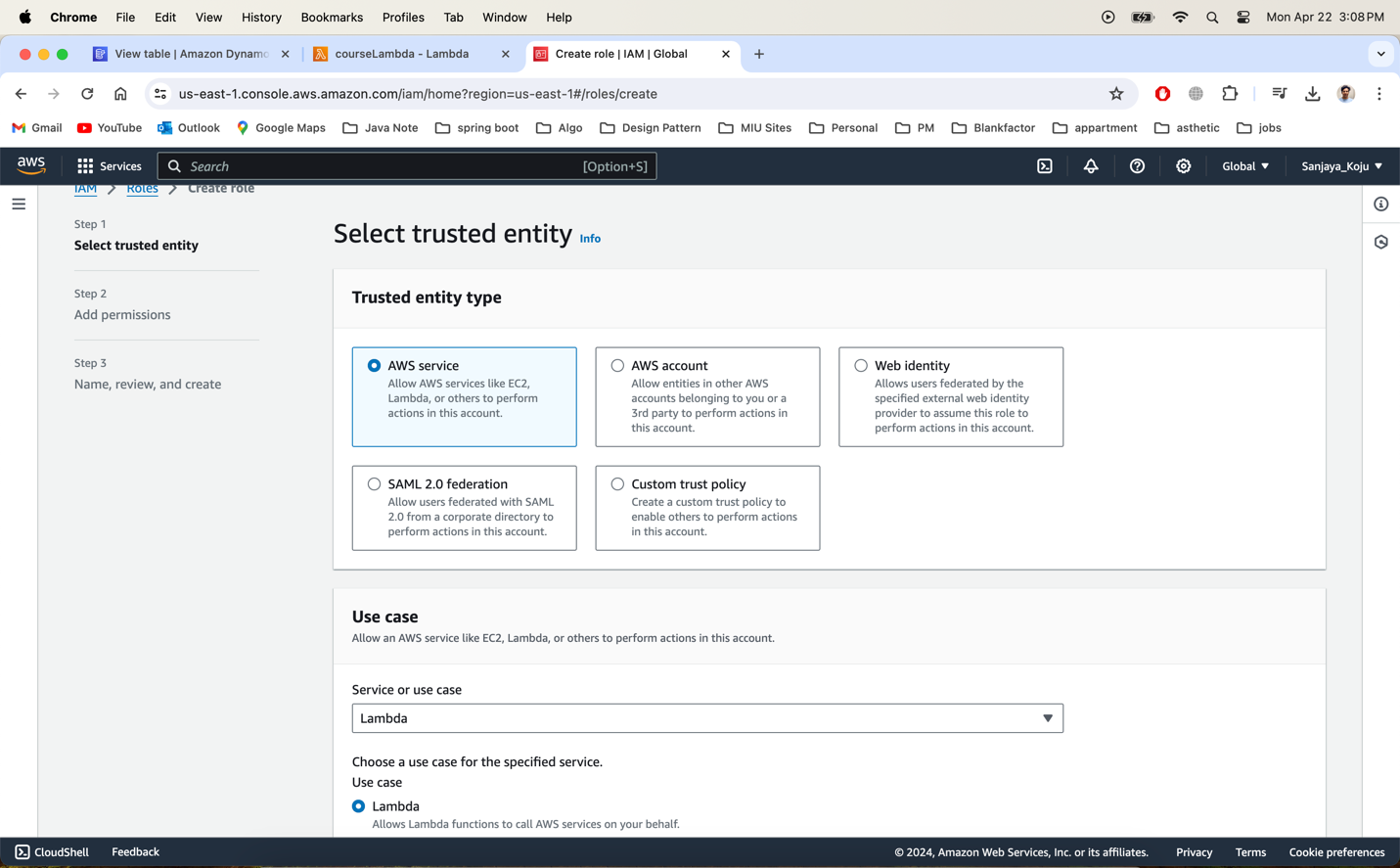
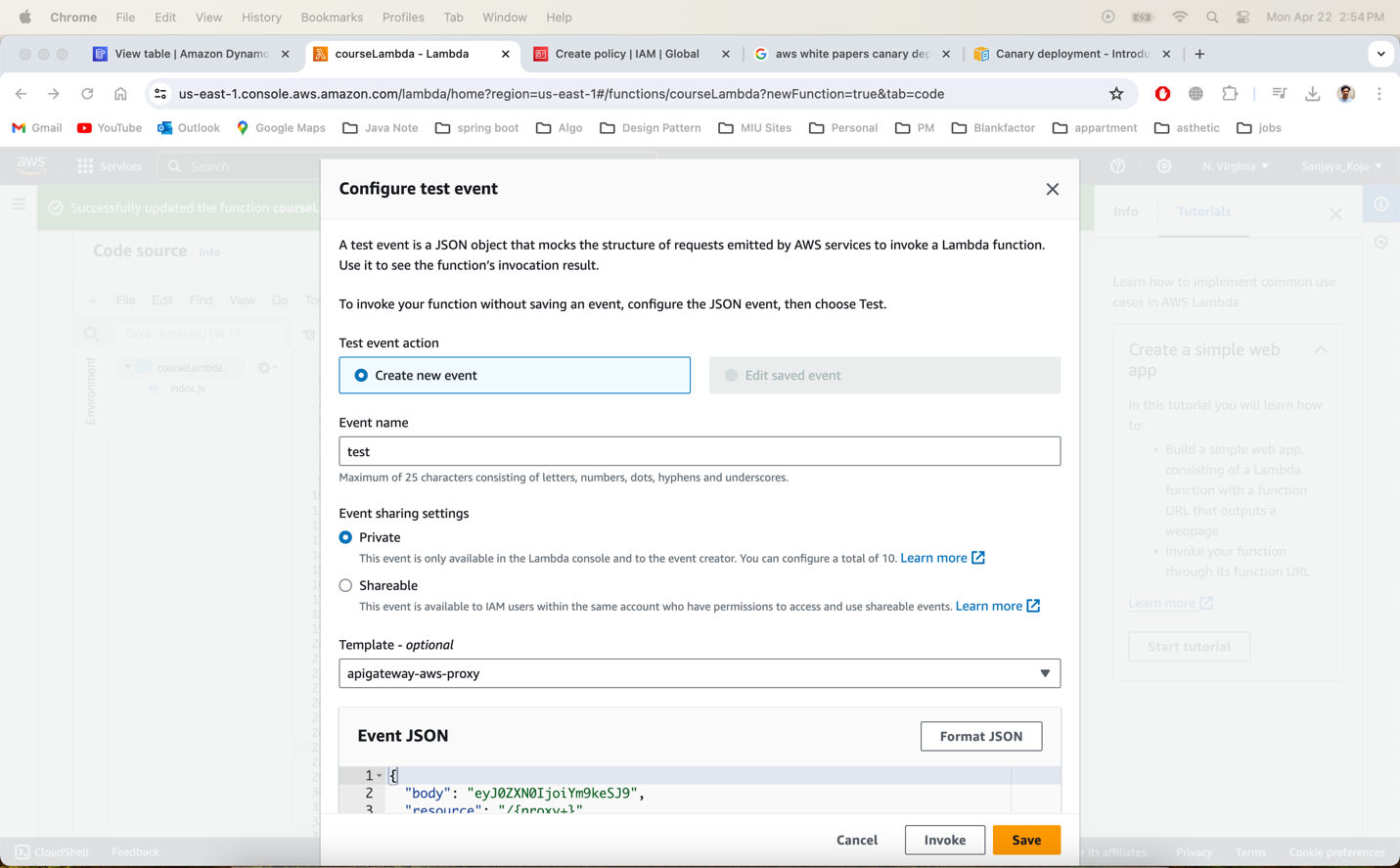
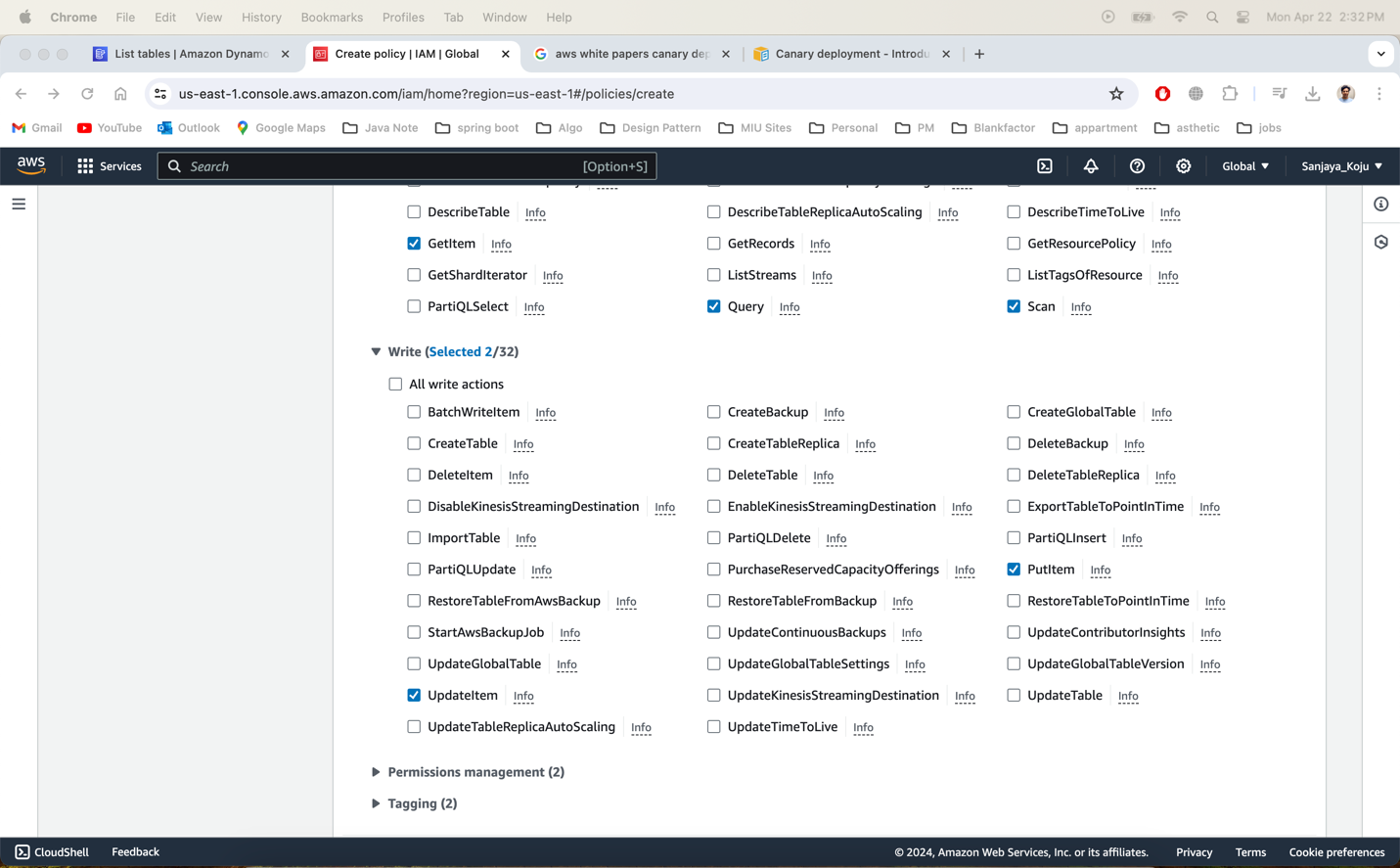
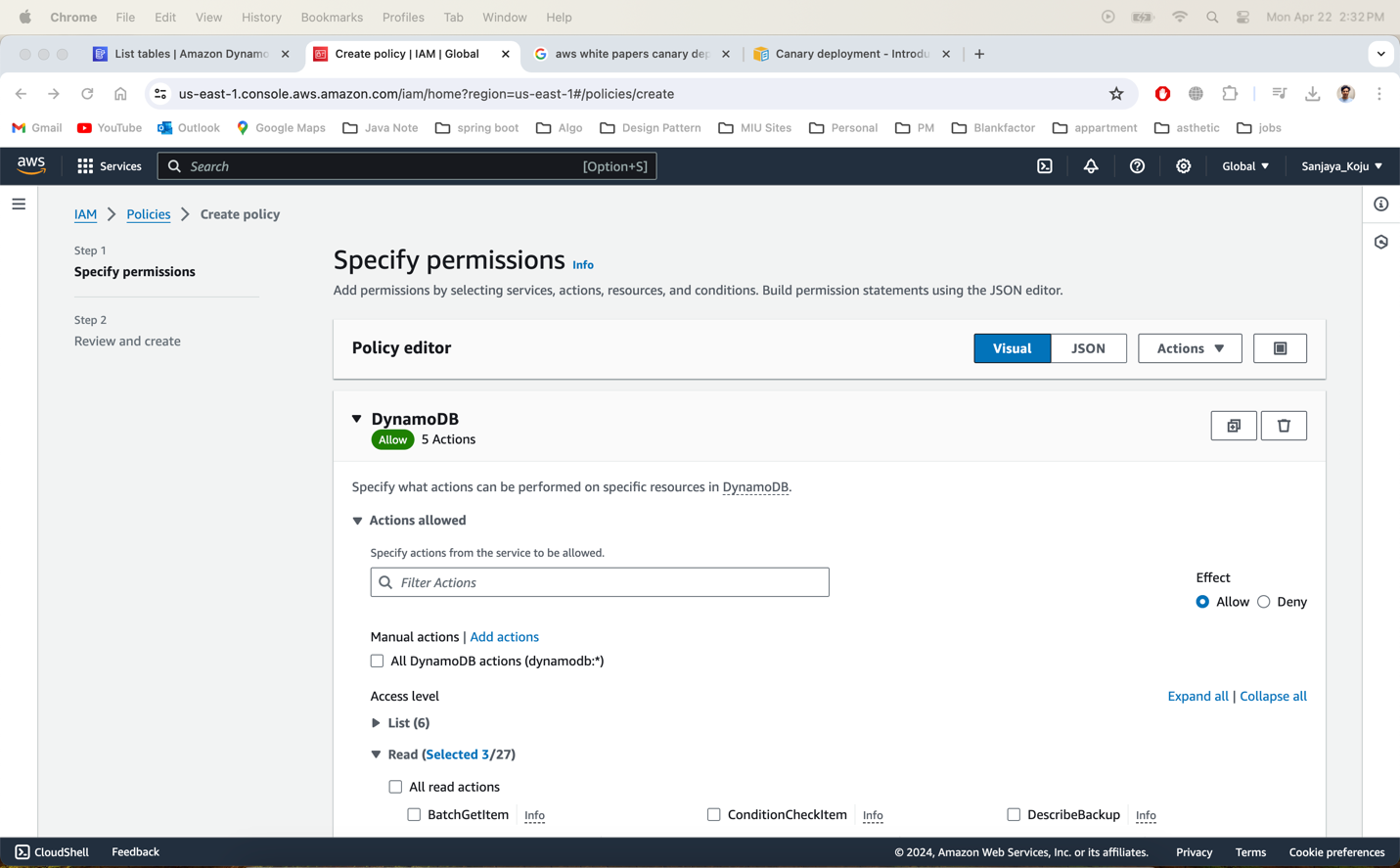
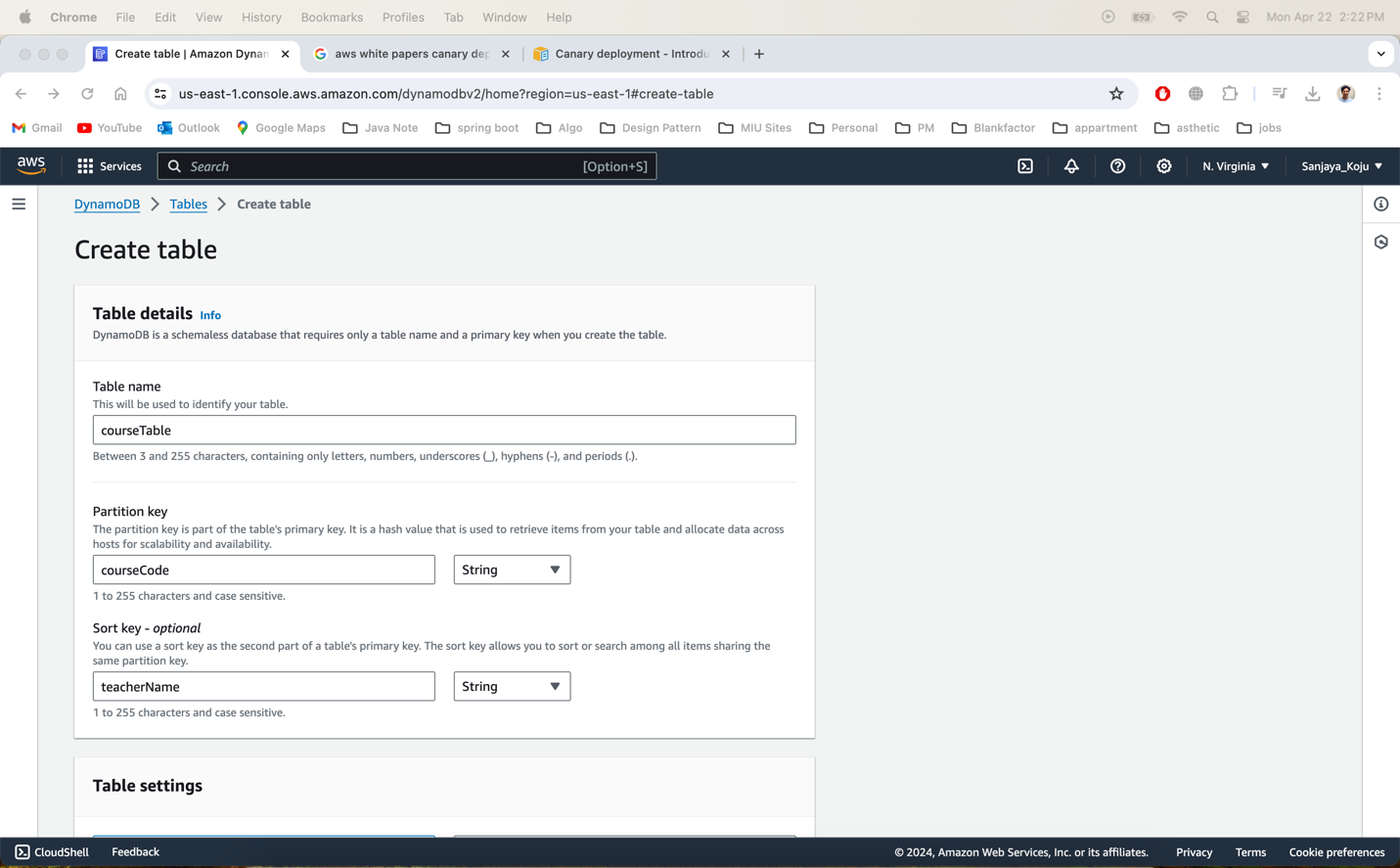
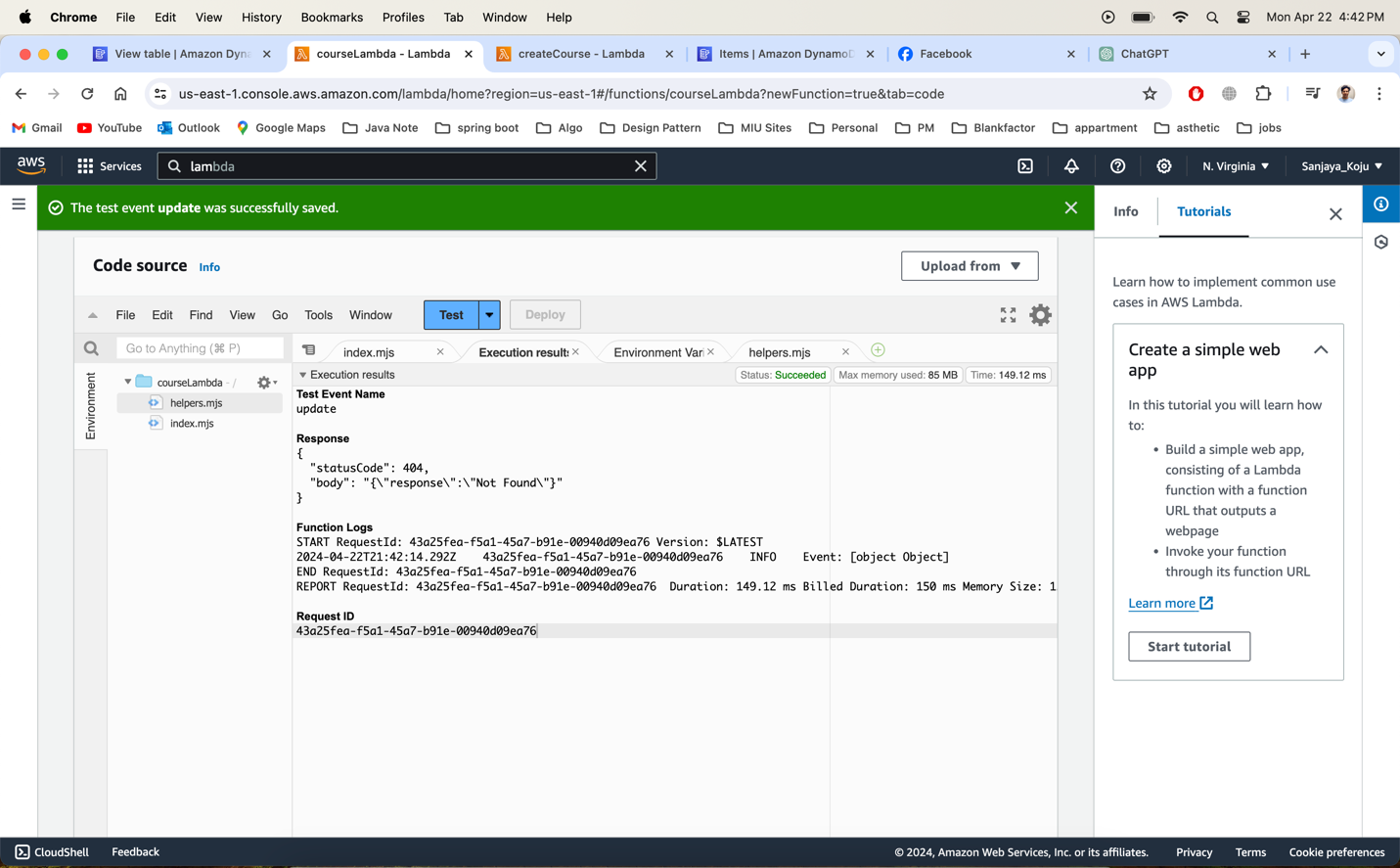
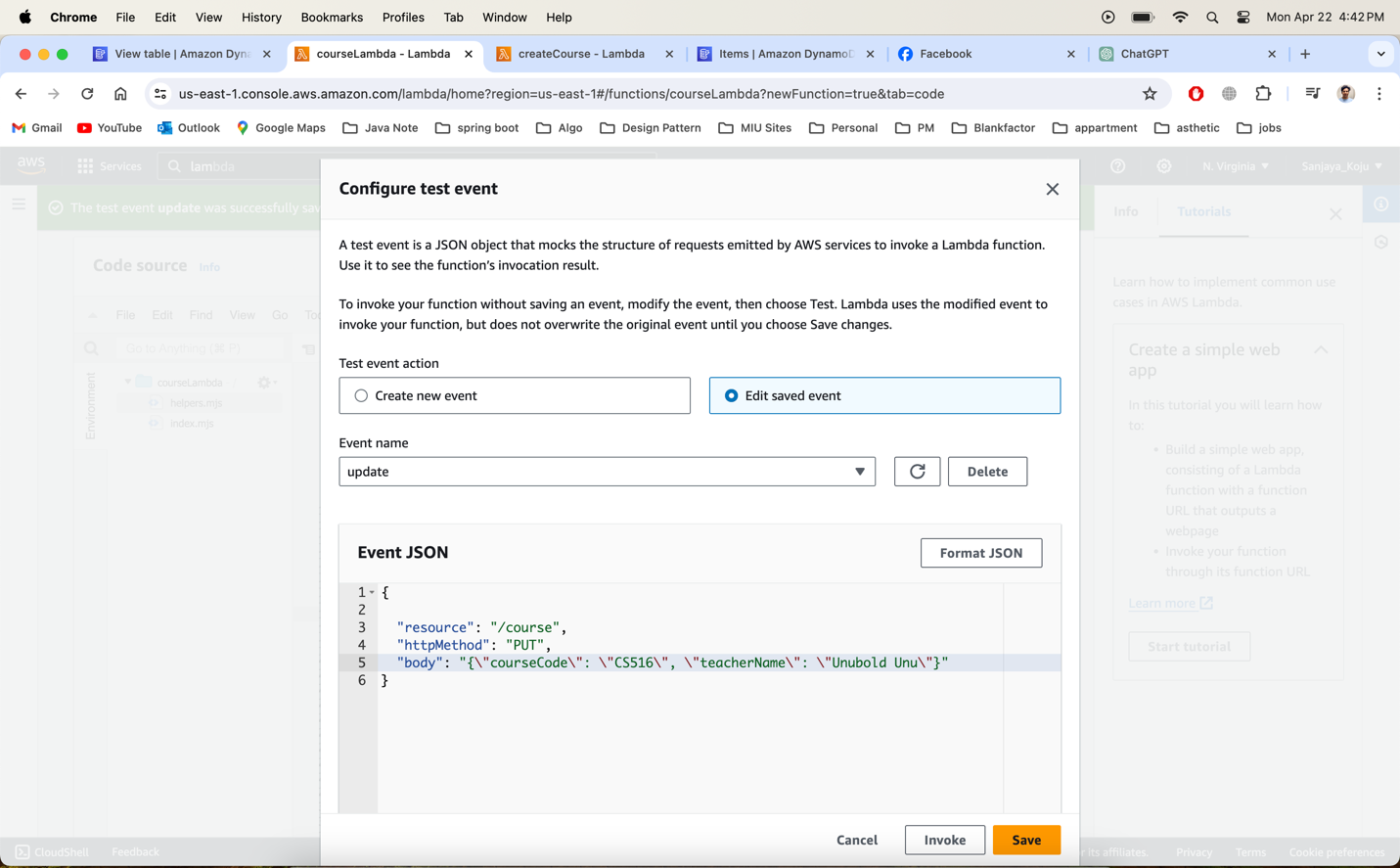
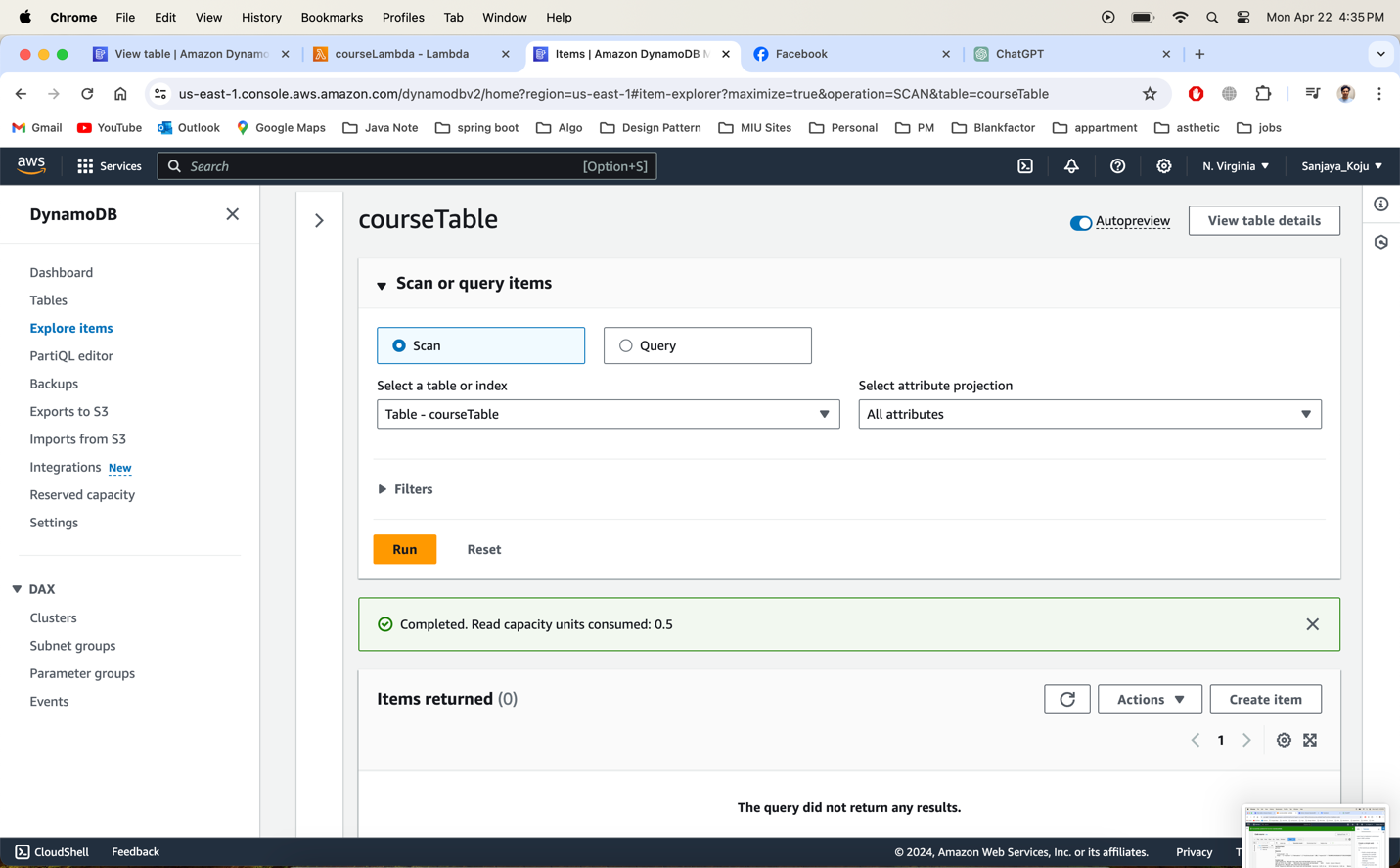
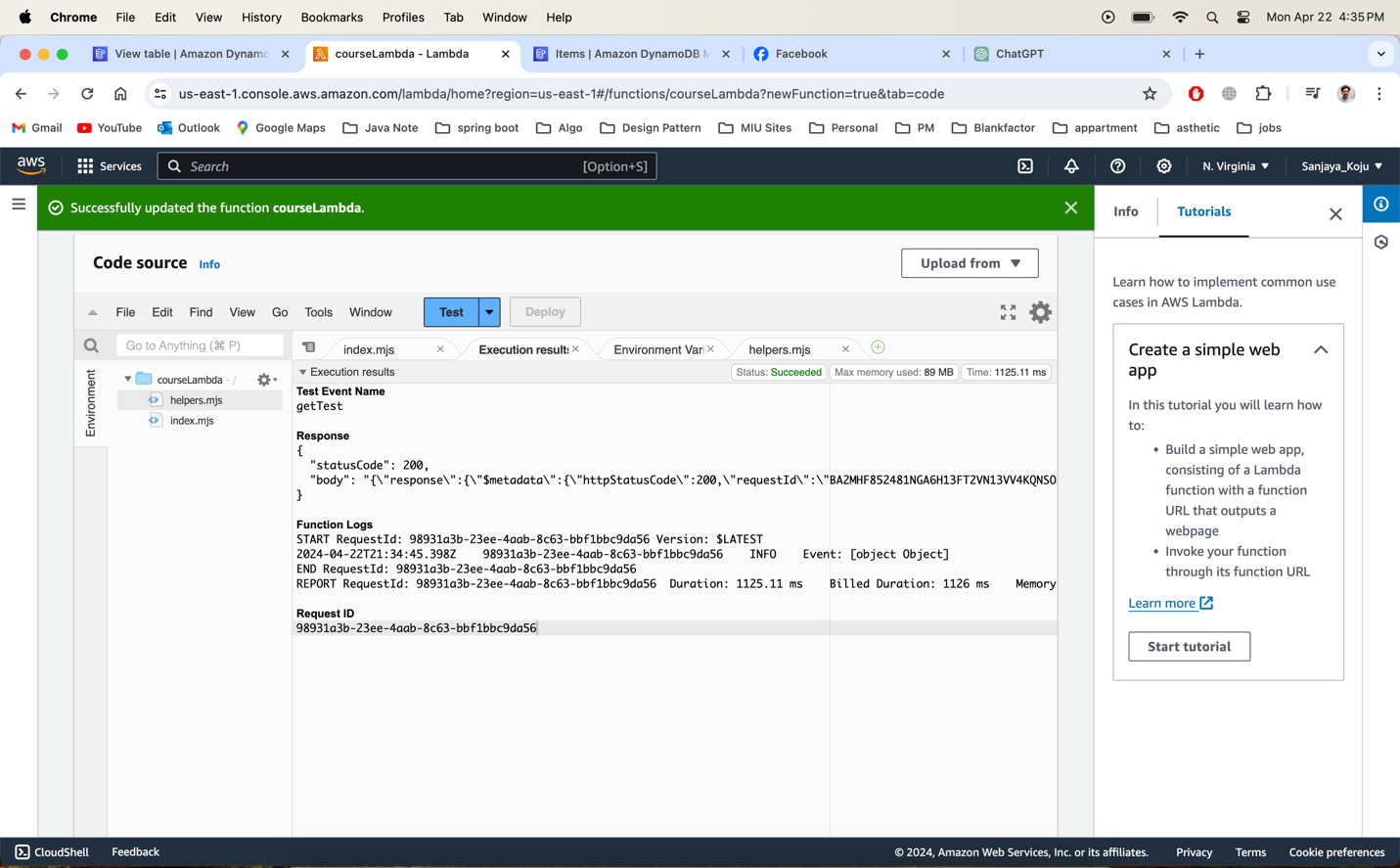
Task 0. Practice version, alias, and weighted (canary) deployment



Task 1. Create a DynamoDB table, Lambda, and implement the Save functionality



Task 2 – Implement the rest of the CRUD operations



**Ignored this part**

**Codes for future references:**

Task 1:

const AWS = require("aws-sdk");

const dynamodb = new AWS.DynamoDB({

apiVersion: "2012-08-10"

});

const tableName = "courseTable";

exports.handler = async(event) => {

console.log("Request received: "+ JSON.stringify(event));

const saveParameters = {

TableName: tableName,

Item: {

"courseCode": {

S: 'CS516'

},

"courseName": {

S: 'Cloud Computing'

},

"teacherName": {

S: 'Unubold'

},

"students": {

SS: [

"Sanjaya",

"Susan",

"Dina"

]

},

"month": {

N: "7"

},

"year": {

N: "2024"

}

}

};

await dynamodb.putItem(saveParameters).promise();

const response = {

statusCode: 200,

body: JSON.stringify('An Item is Saved.')

};

return response;

};