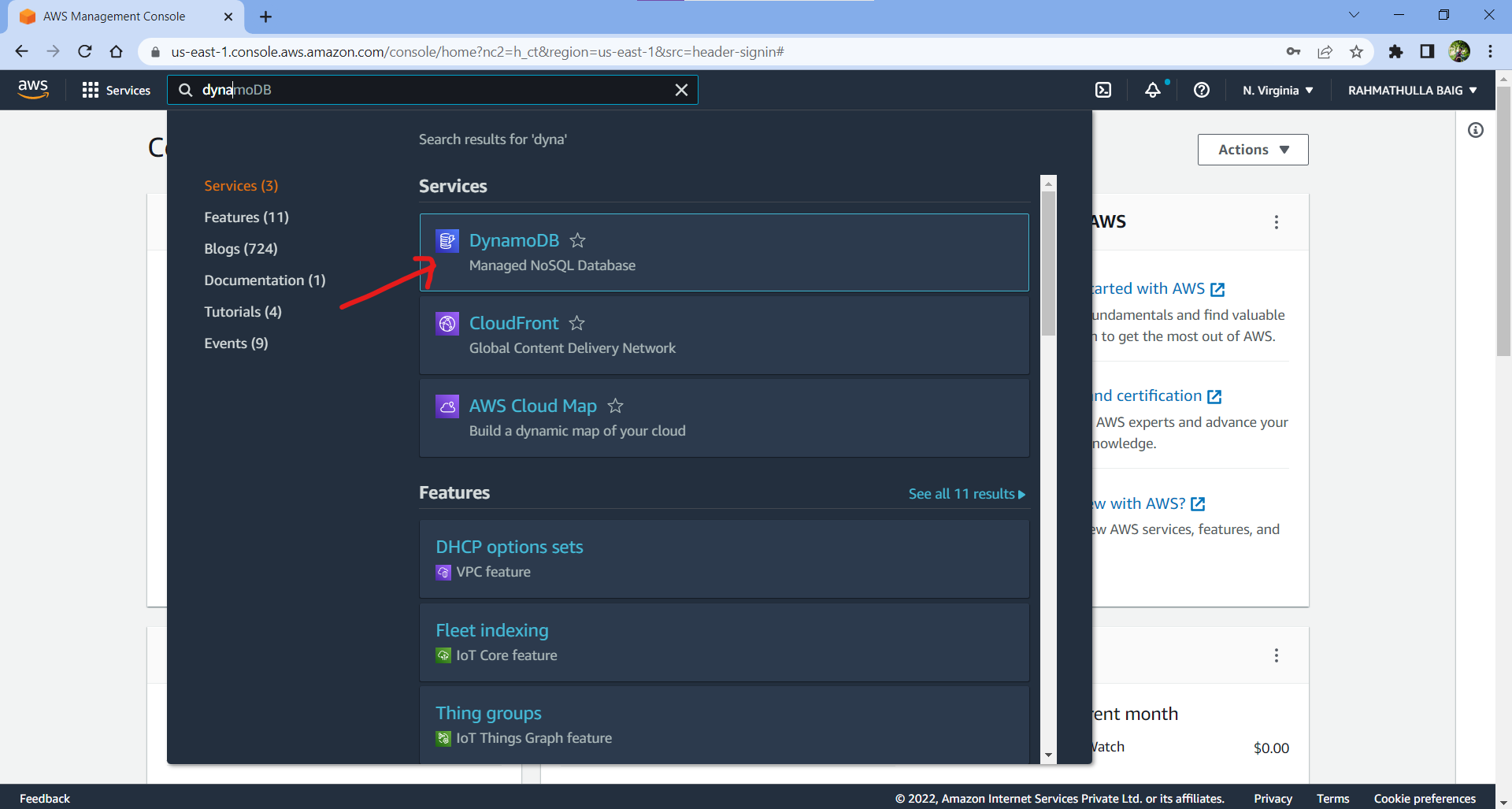
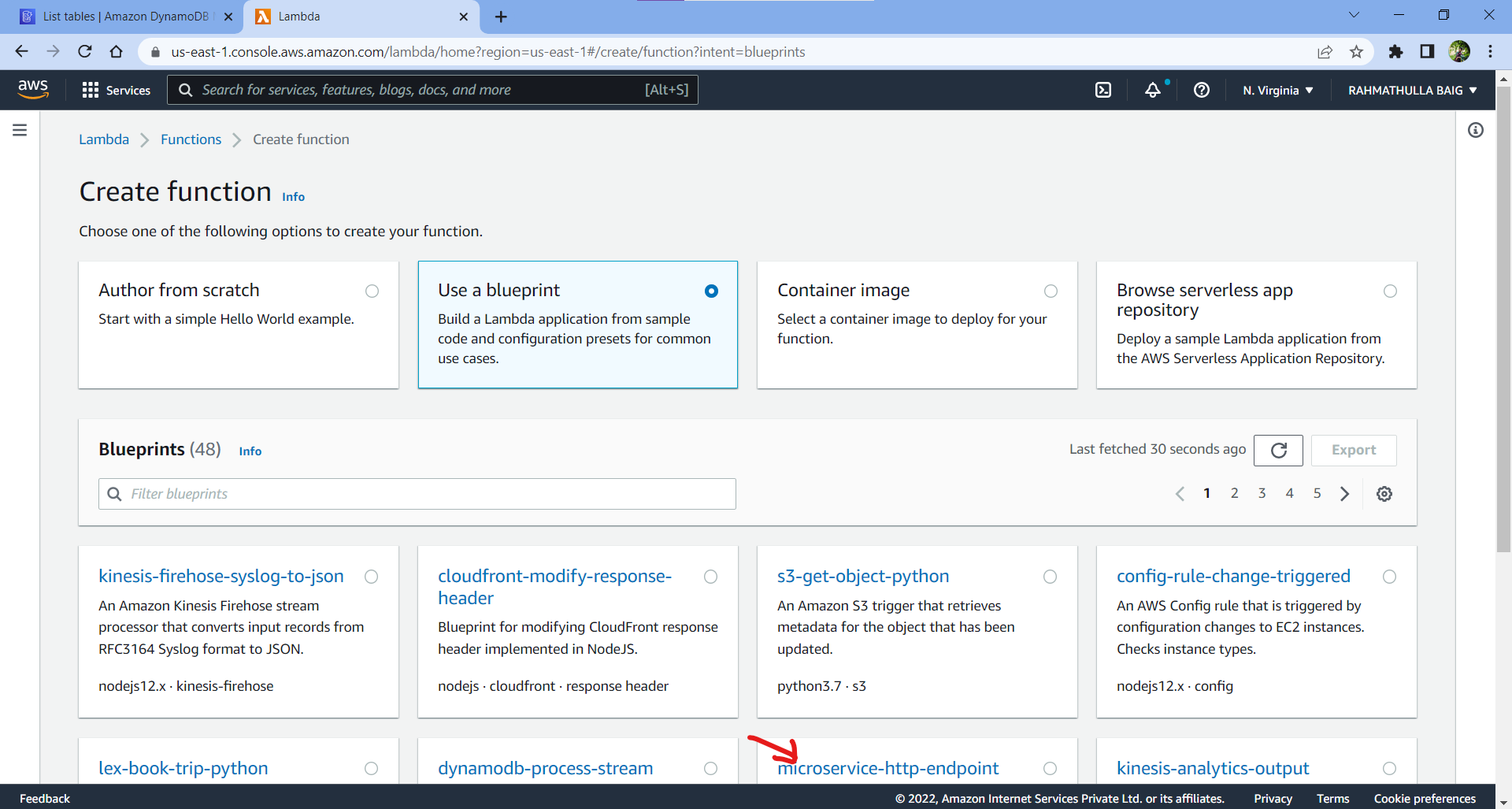
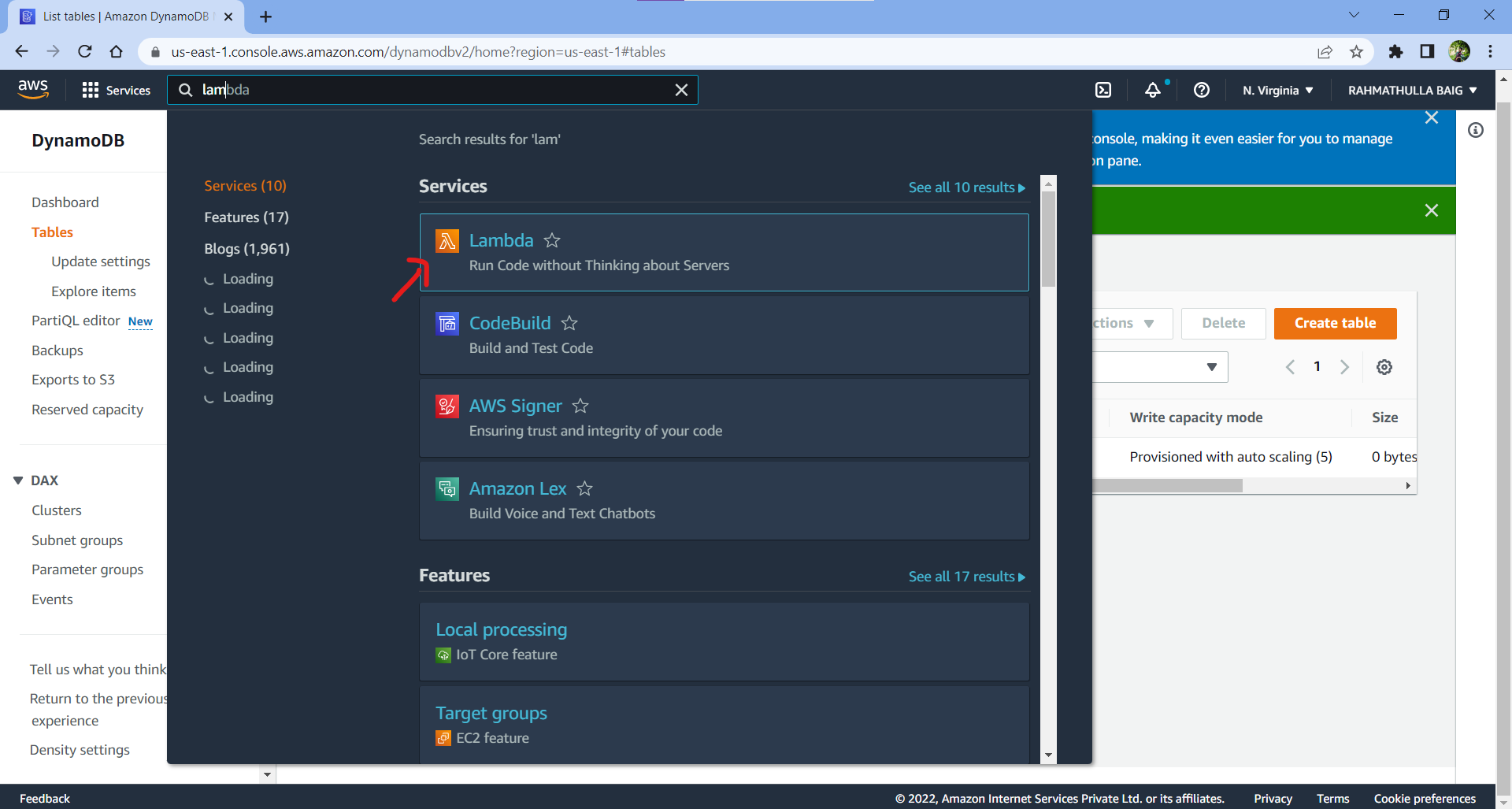
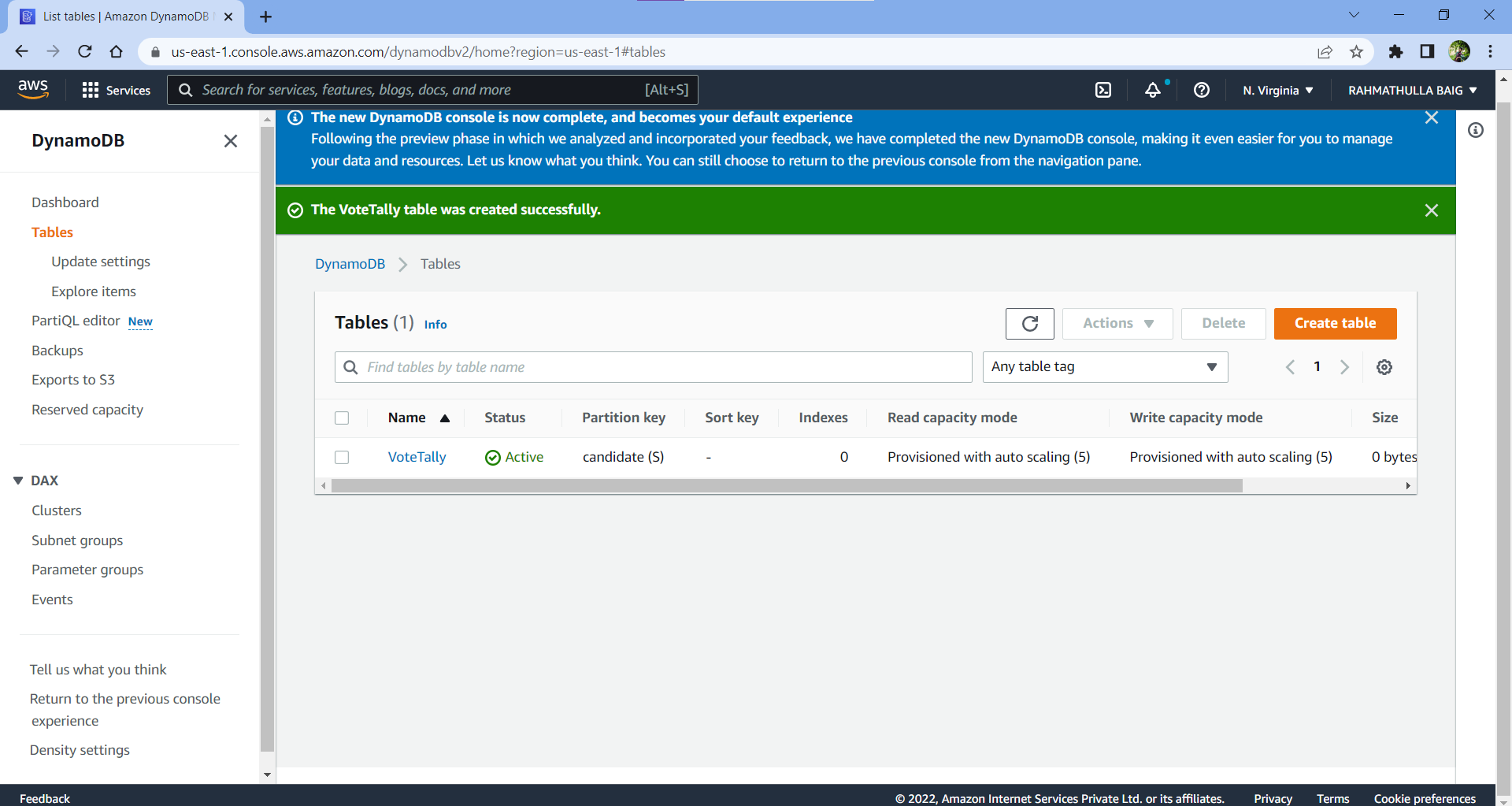
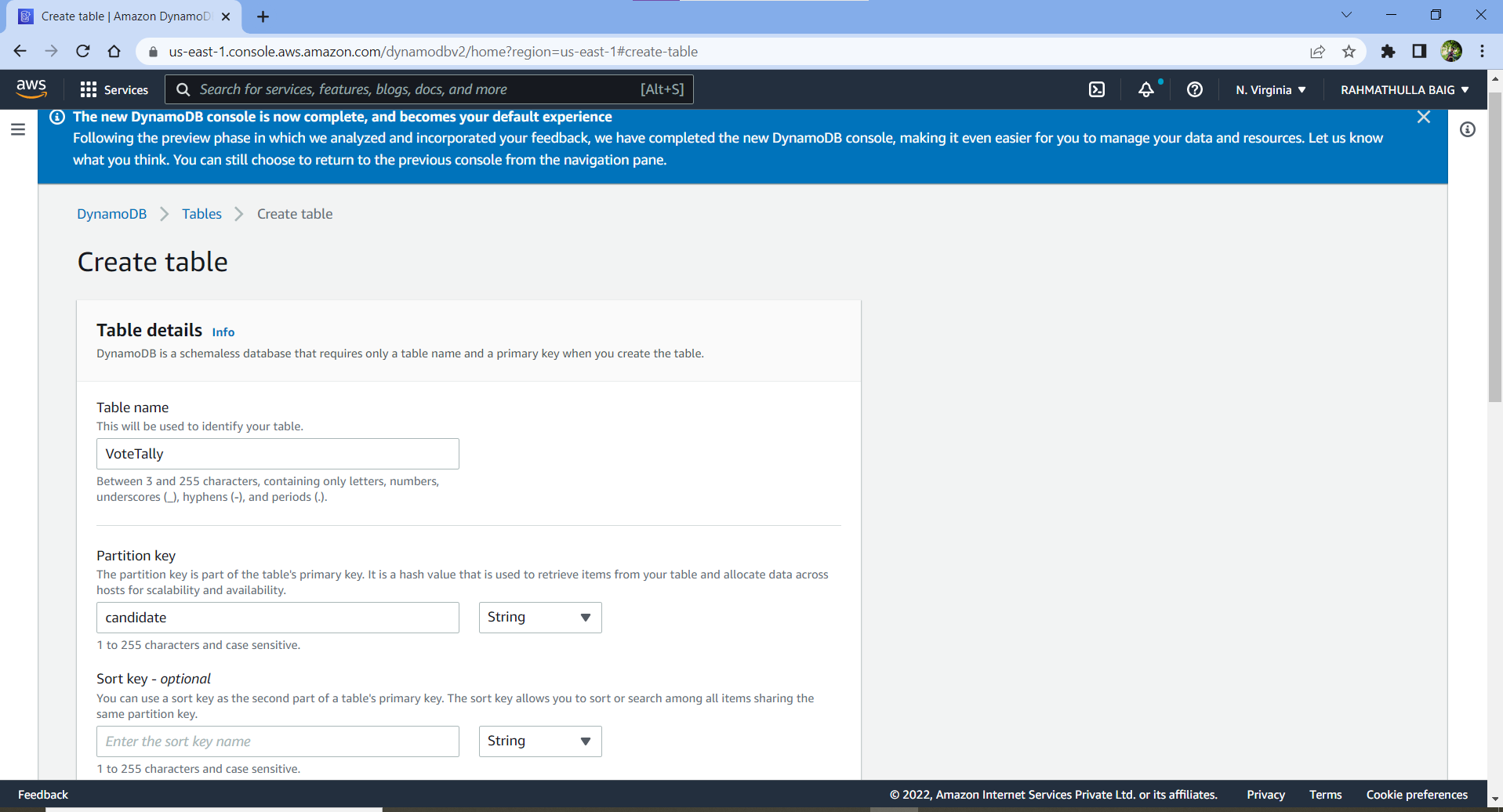
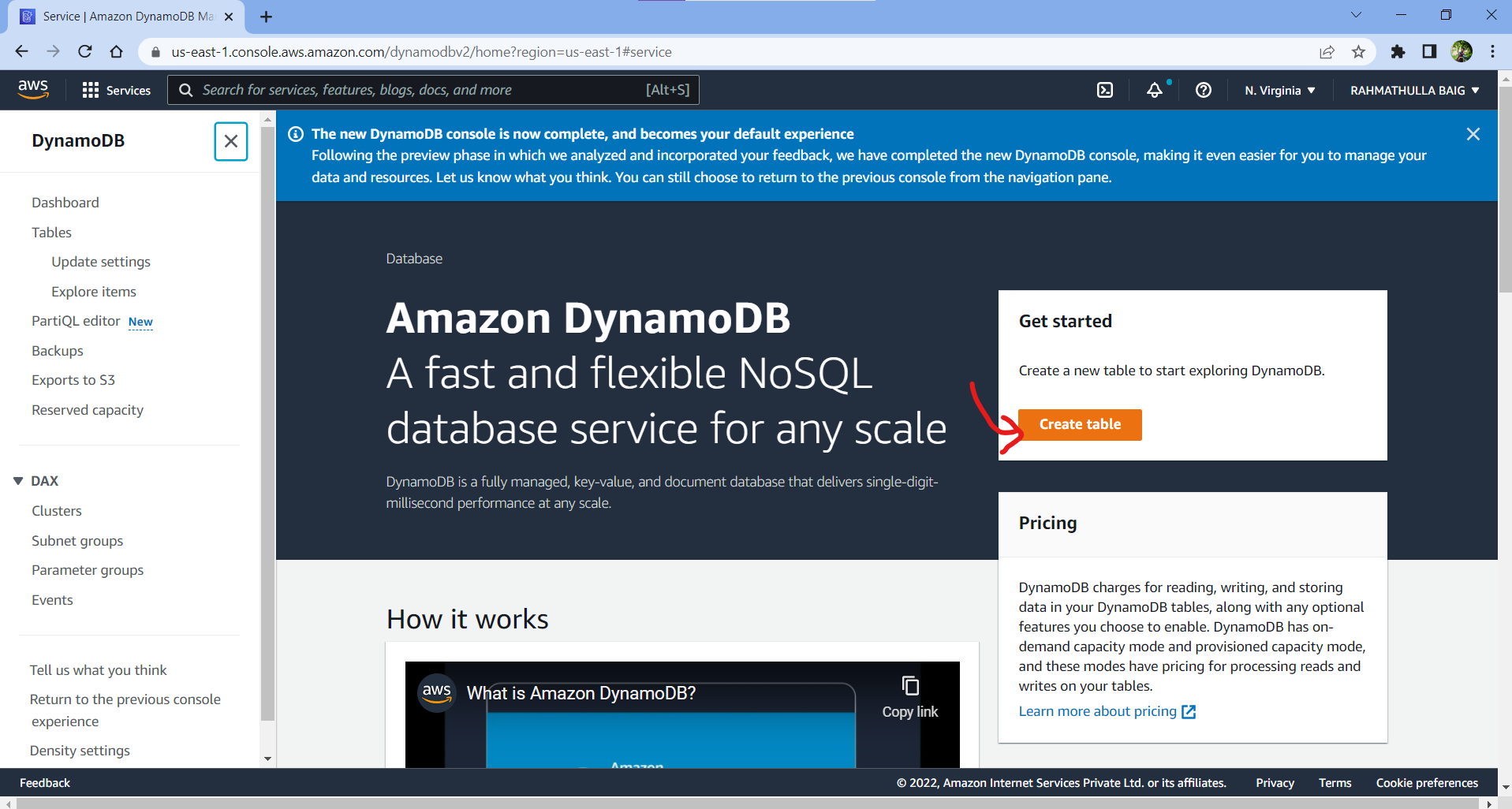
Voting Application Using AWS Services

190030505

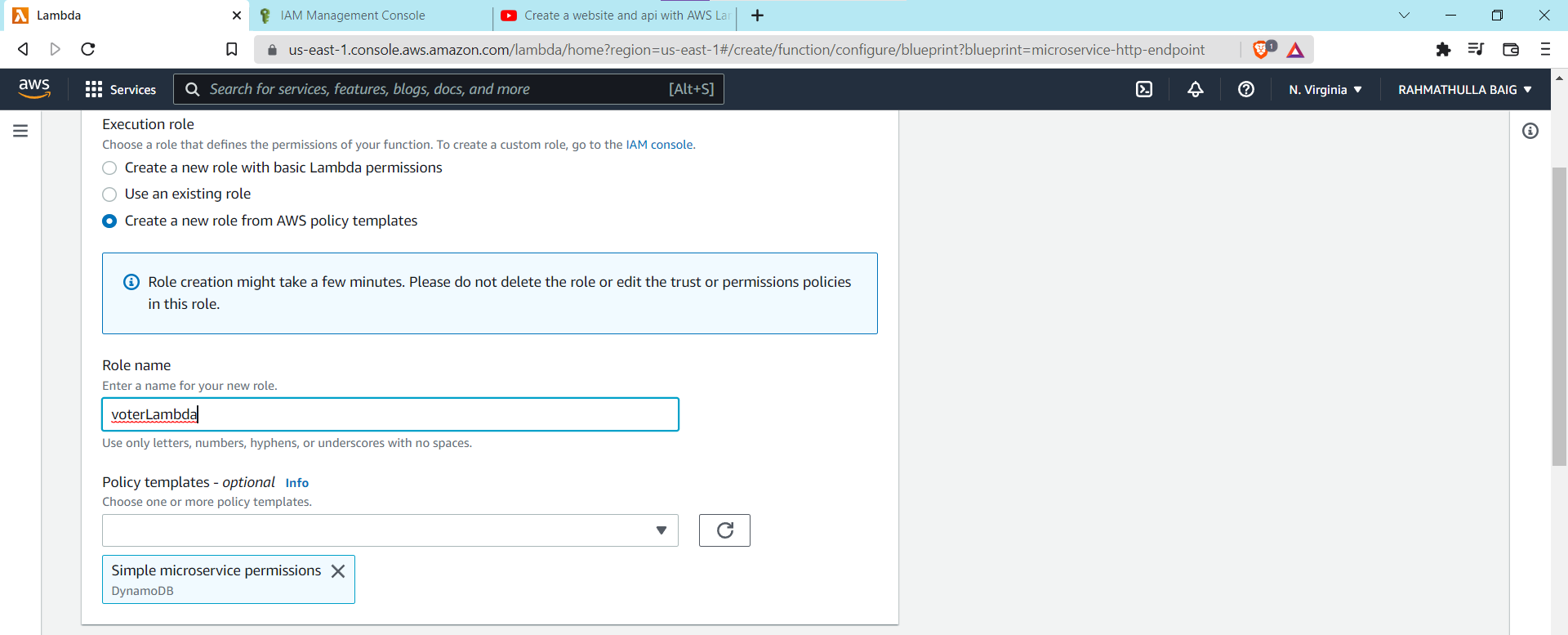
Goda.MidhunKumar

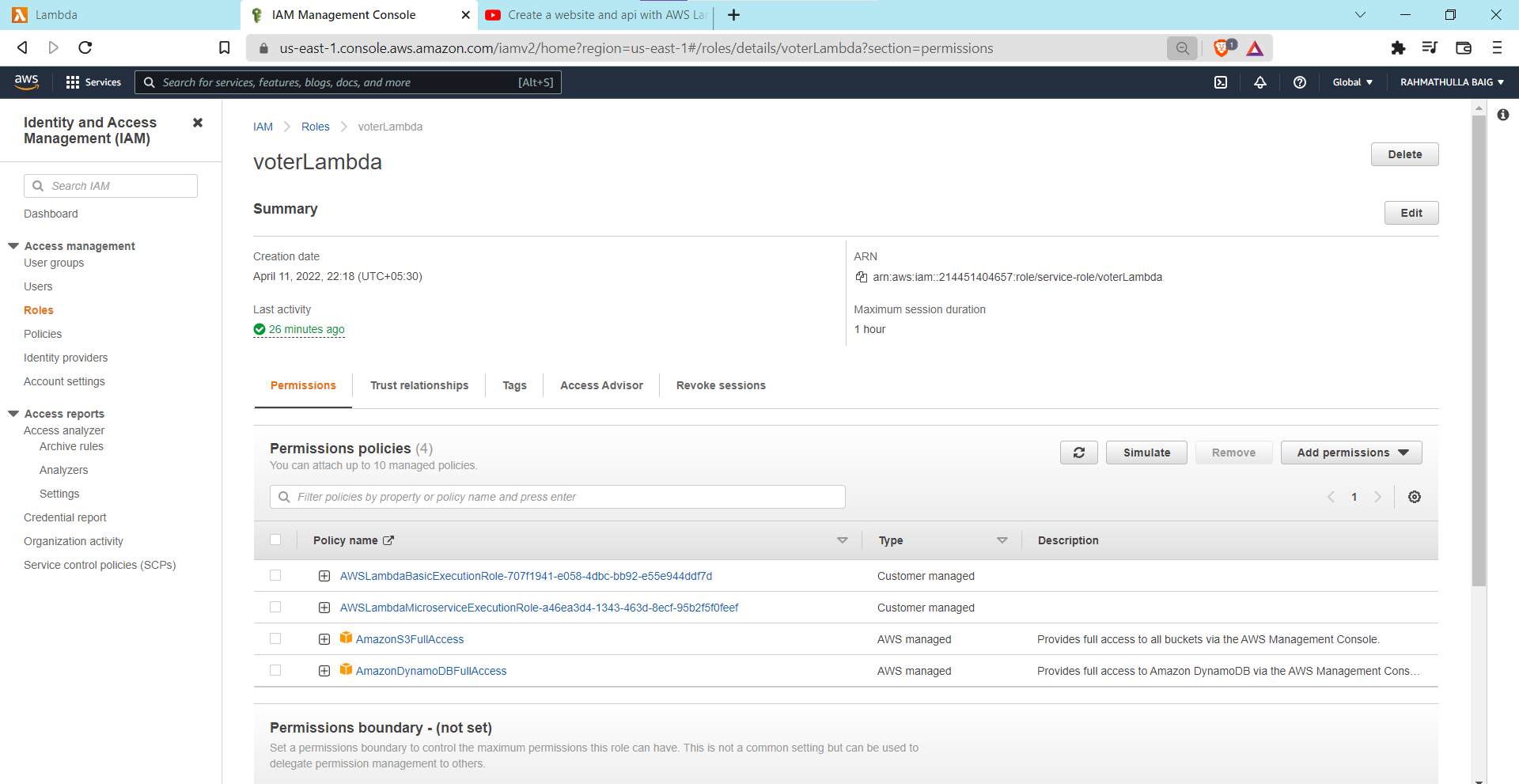


‘

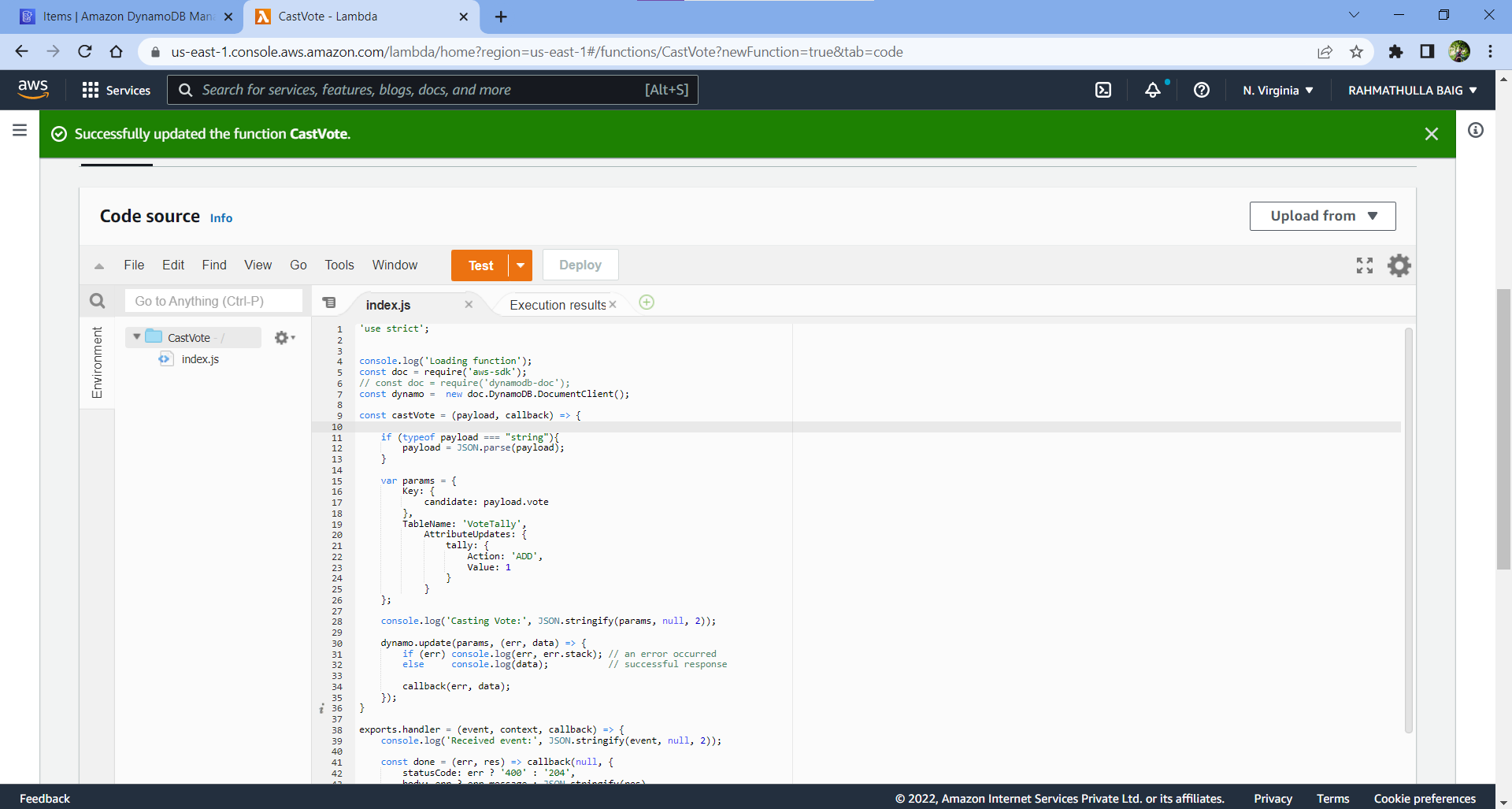
Create Lambda Function (NodeJS)

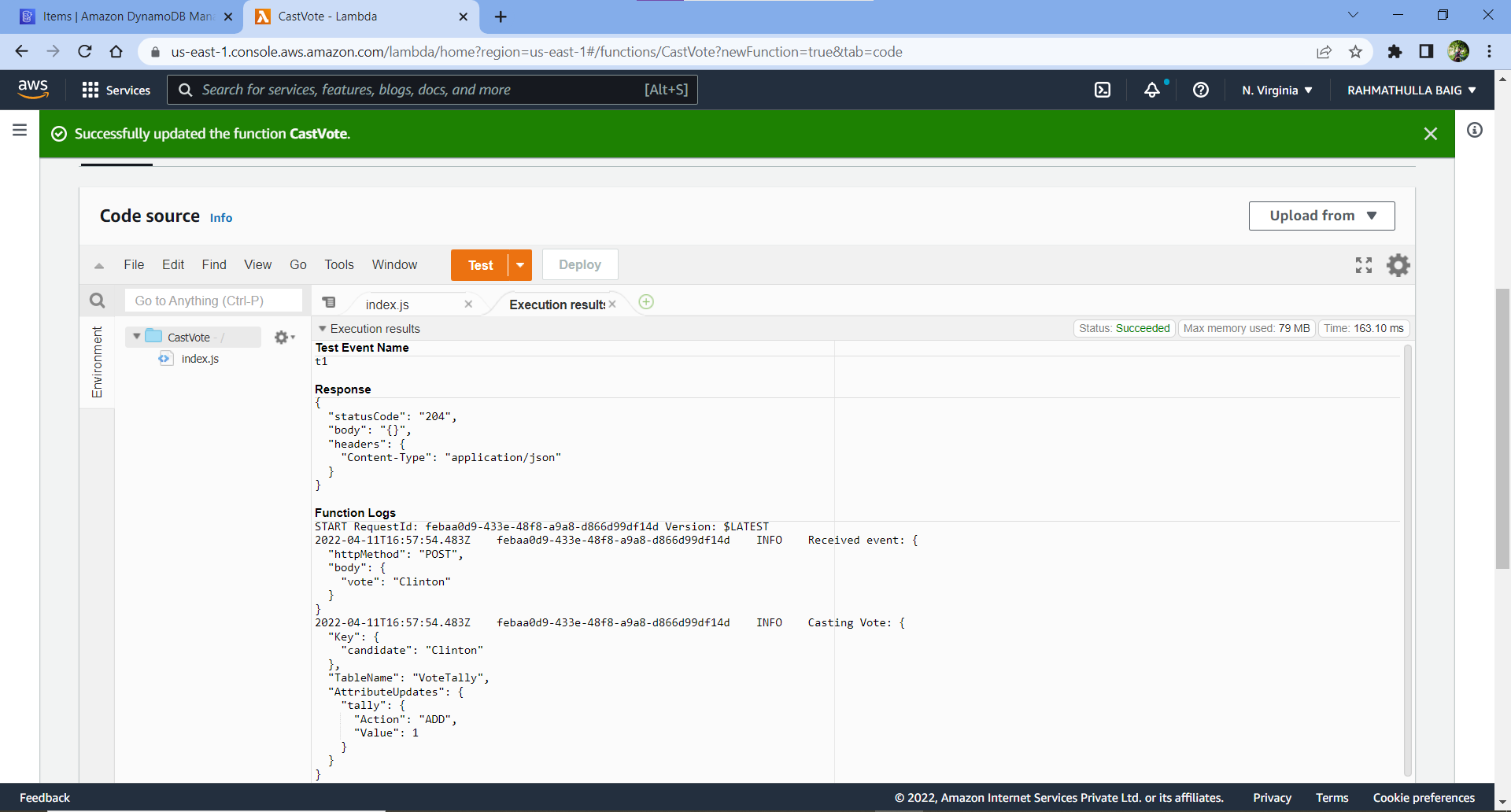
Create a New Role





Which is in files CastVote.js -> click on test





Lambda Code -> CastVote.js

'use strict';

console.log('Loading function');

const doc = require('aws-sdk');

// const doc = require('dynamodb-doc');

const dynamo = new doc.DynamoDB.DocumentClient();

const castVote = (payload, callback) => {

if (typeof payload === "string"){

payload = JSON.parse(payload);

}

var params = {

Key: {

candidate: payload.vote

},

TableName: 'VoteTally',

AttributeUpdates: {

tally: {

Action: 'ADD',

Value: 1

}

}

};

console.log('Casting Vote:', JSON.stringify(params, null, 2));

dynamo.update(params, (err, data) => {

if (err) console.log(err, err.stack); // an error occurred

else console.log(data); // successful response

callback(err, data);

});

}

exports.handler = (event, context, callback) => {

console.log('Received event:', JSON.stringify(event, null, 2));

const done = (err, res) => callback(null, {

statusCode: err ? '400' : '204',

body: err ? err.message : JSON.stringify(res),

headers: {

'Content-Type': 'application/json',

},

});

switch (event.httpMethod) {

case 'POST':

castVote(event.body, done);

break;

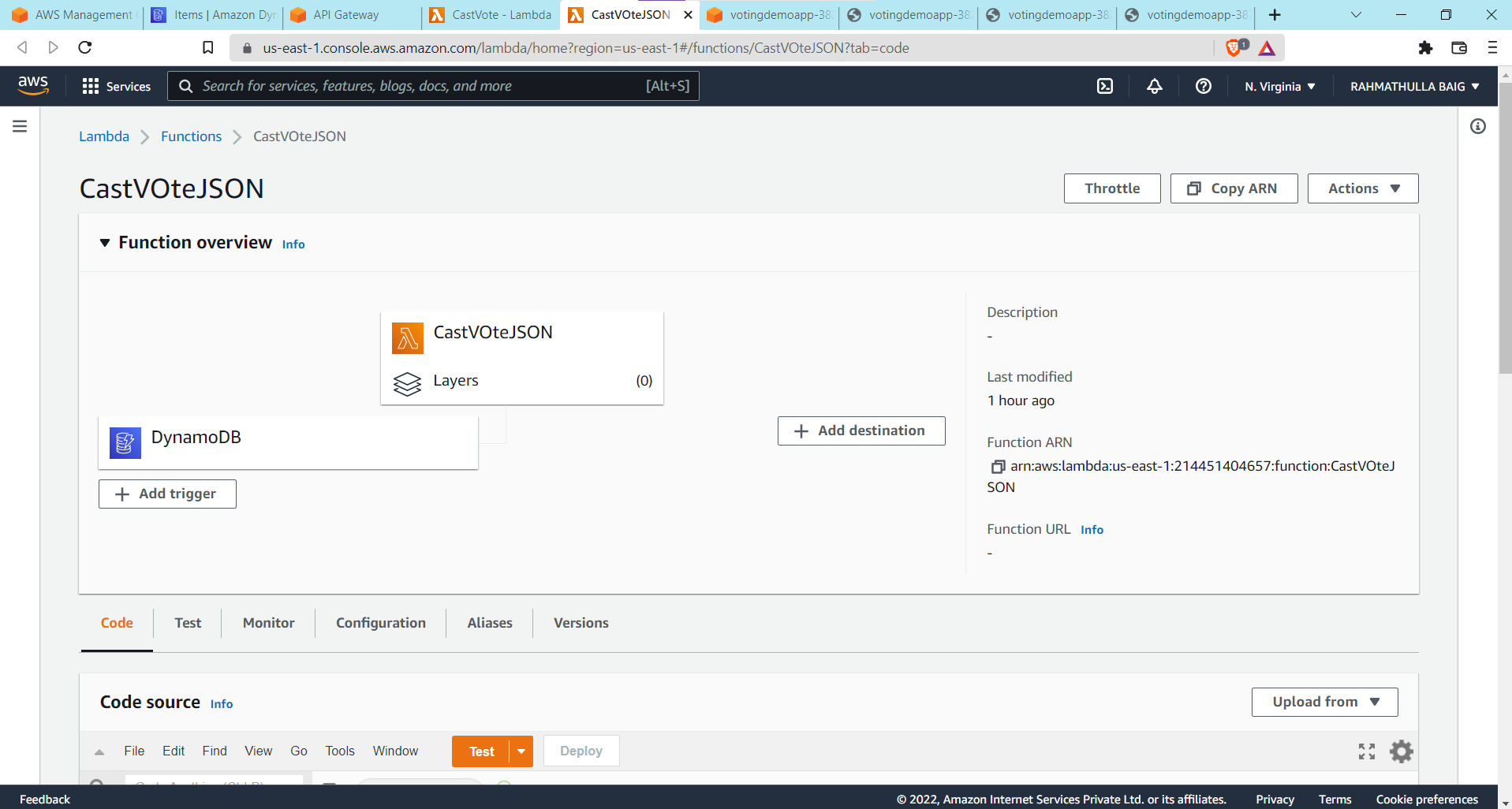
default:

done(new Error(`Unsupported method "${event.httpMethod}"`));

}

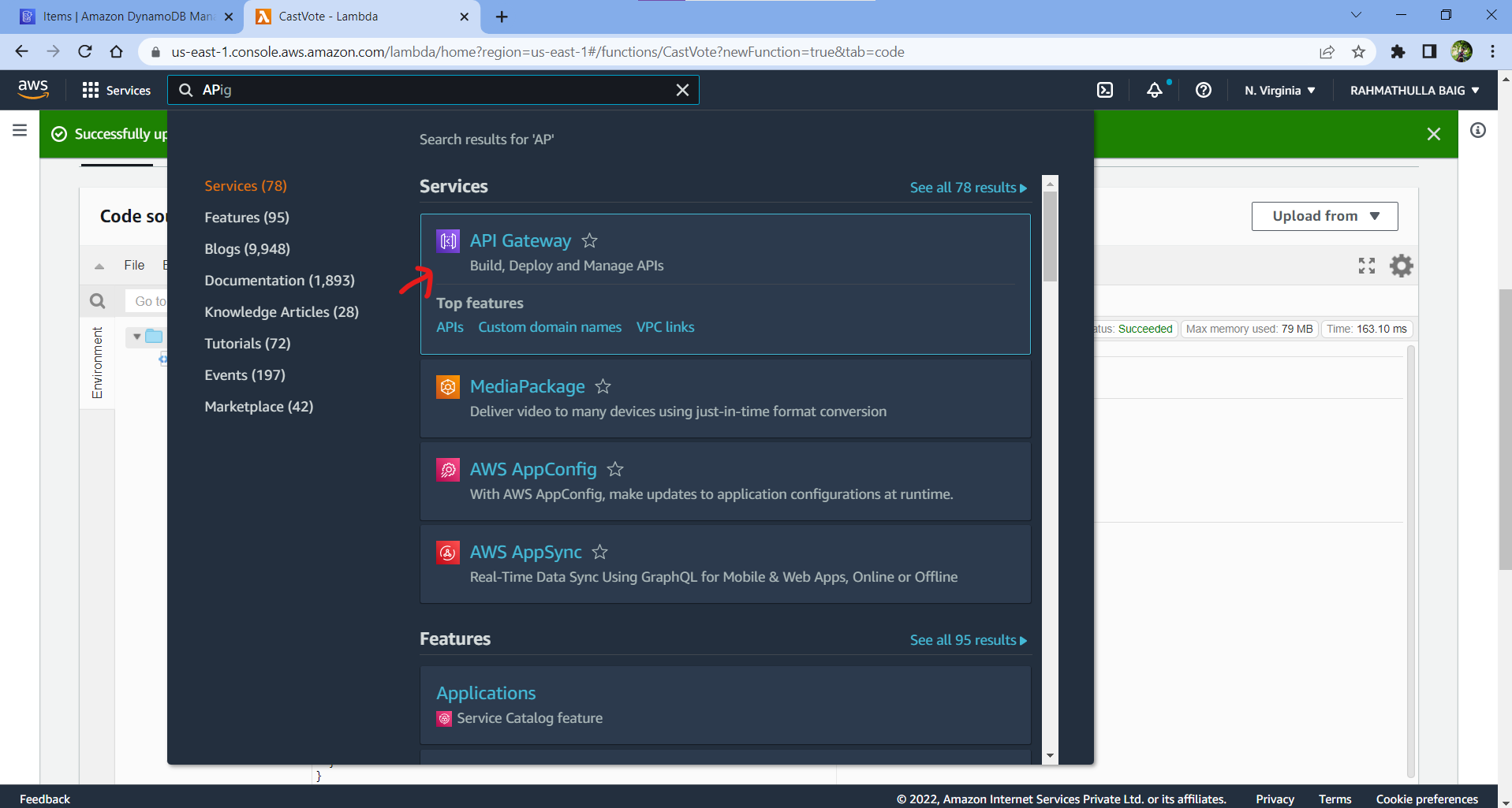
};

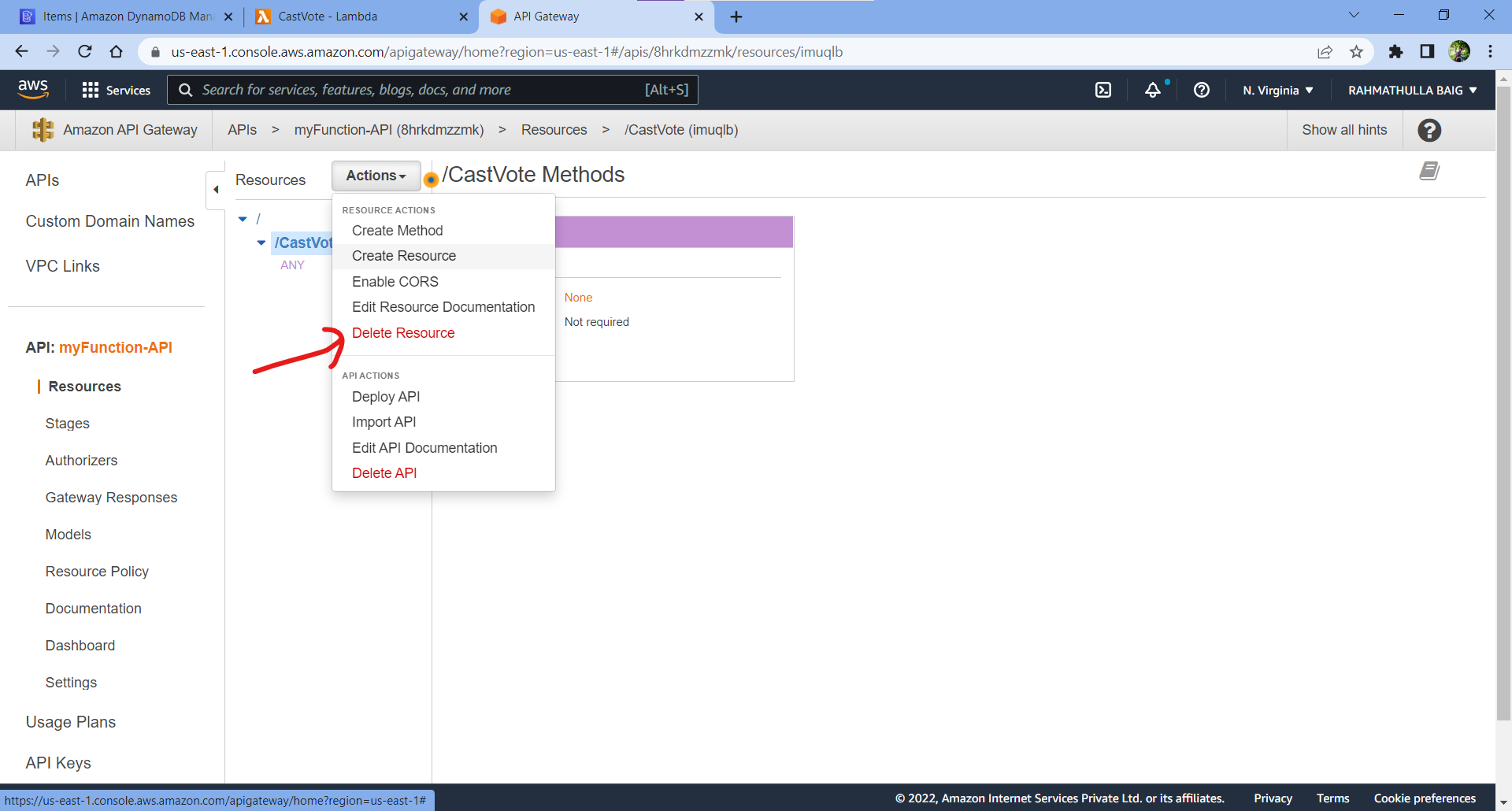
Test->CastVote.js before moving forward

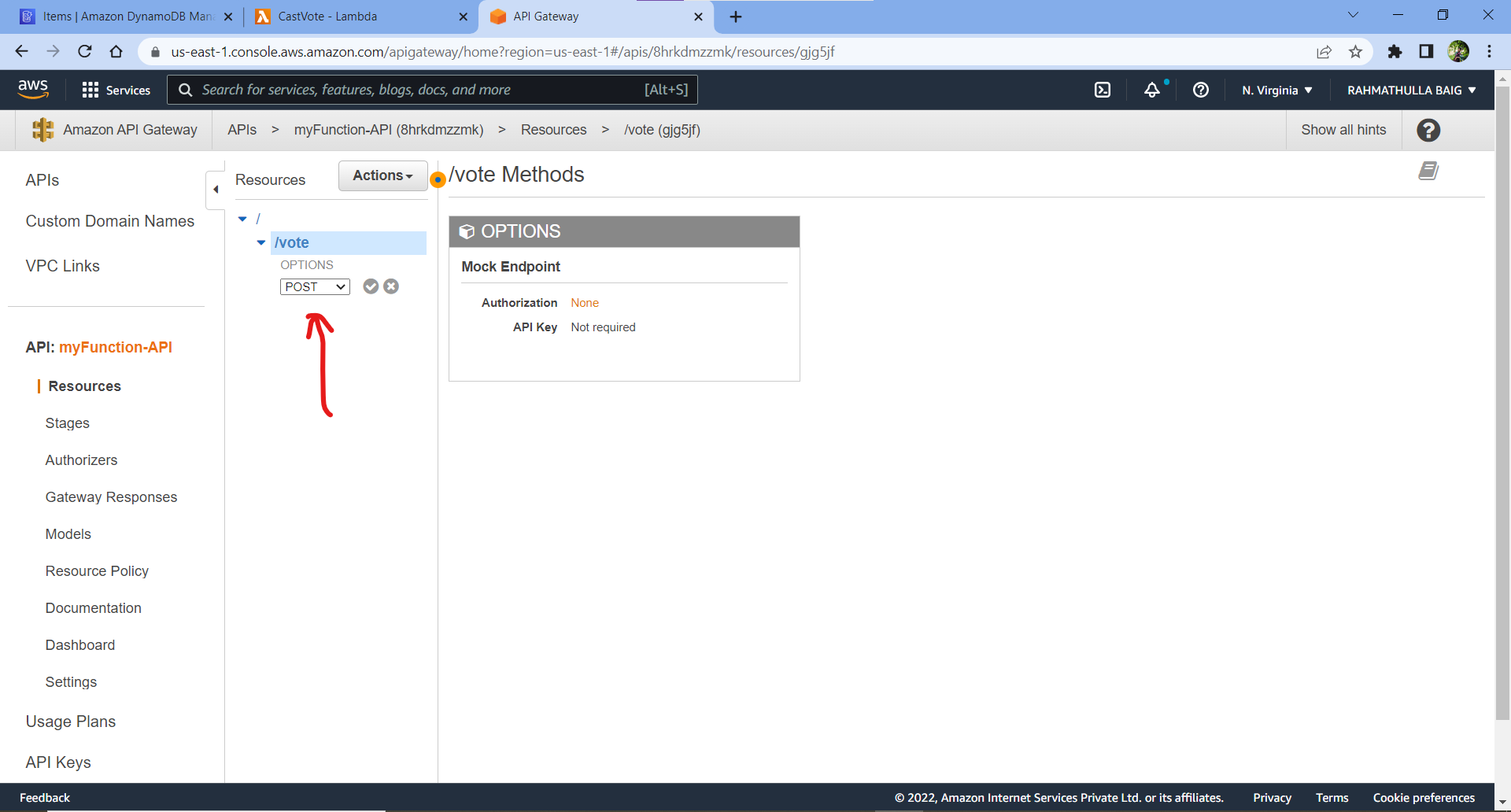
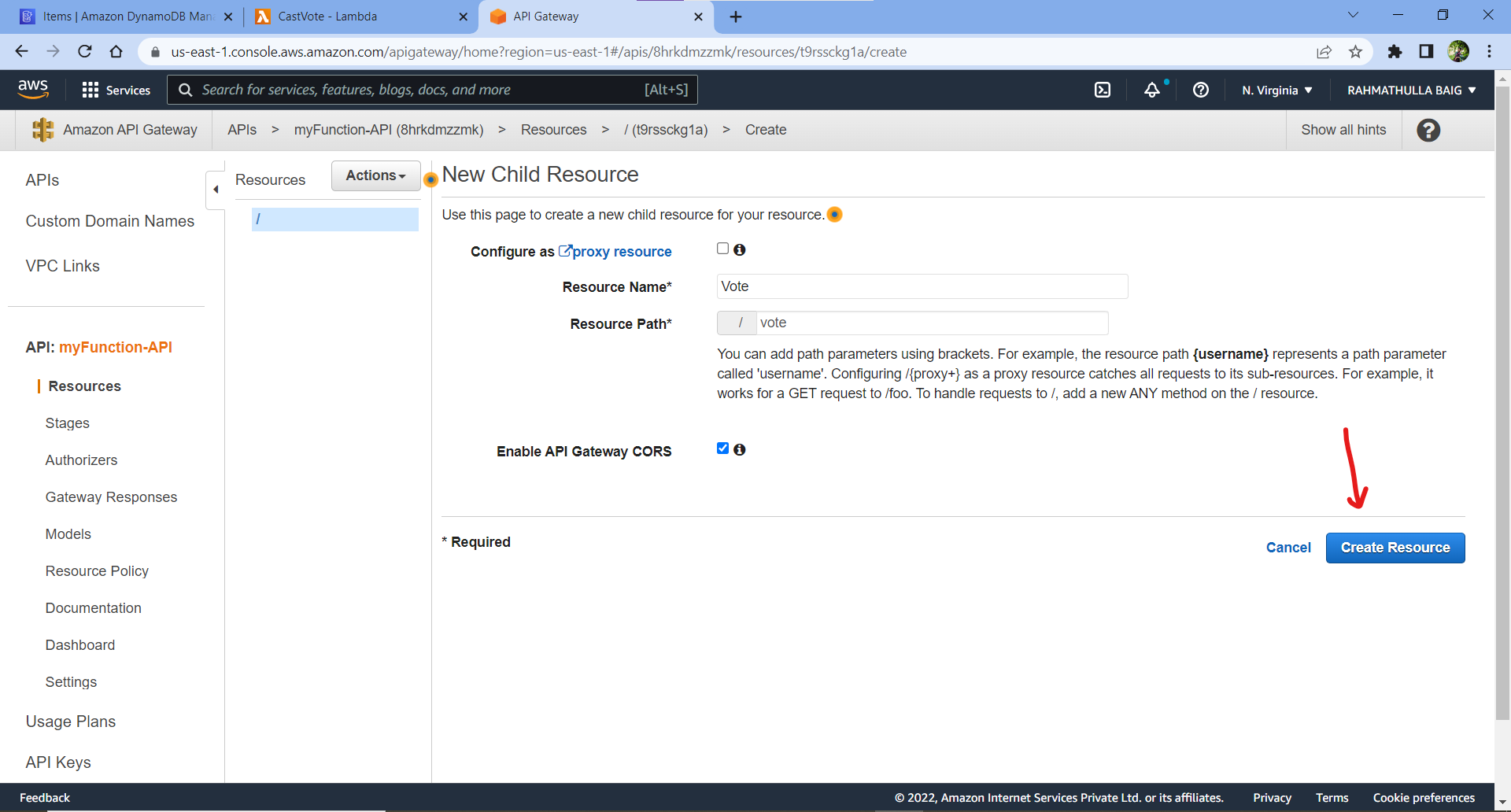


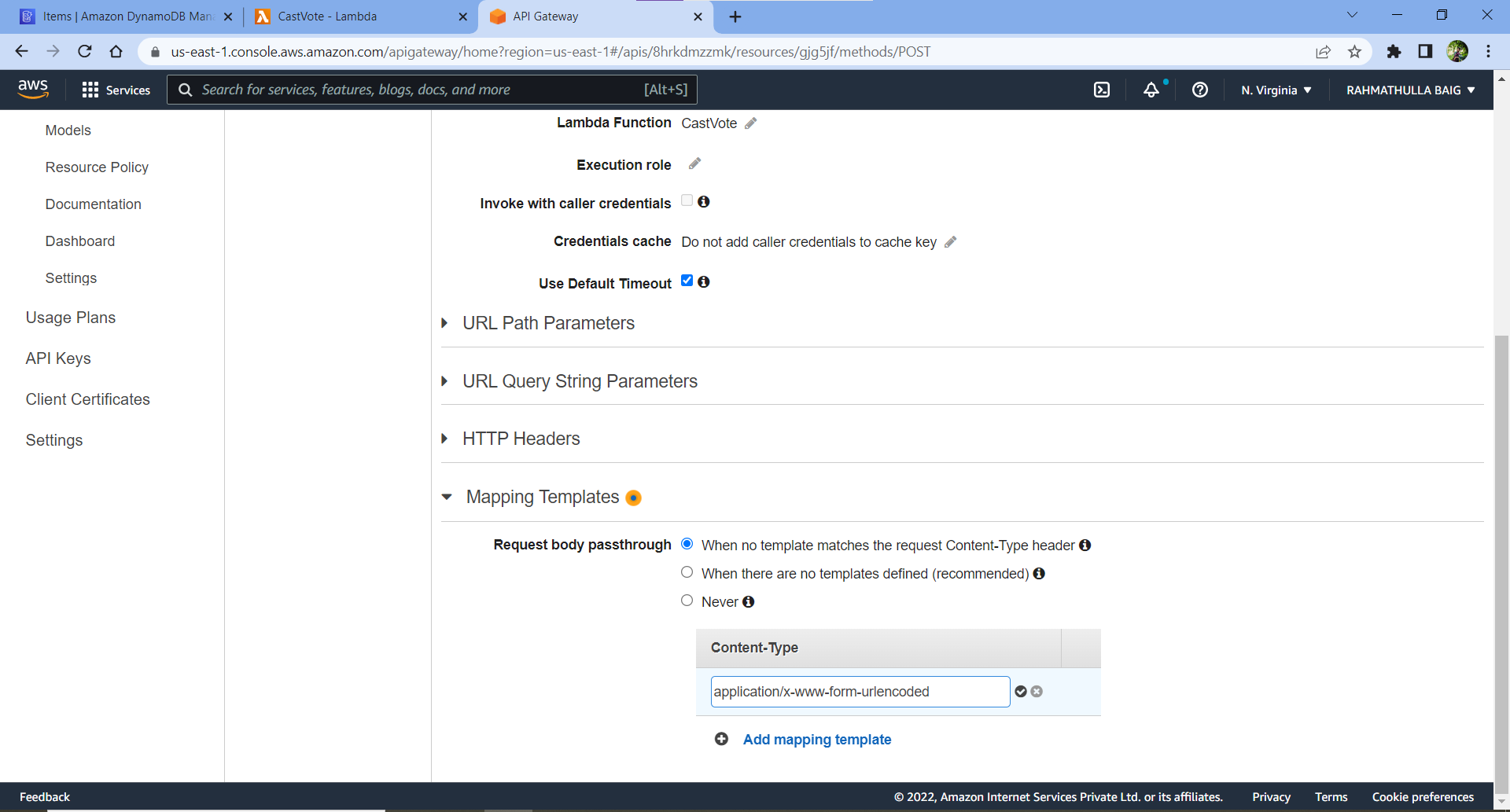
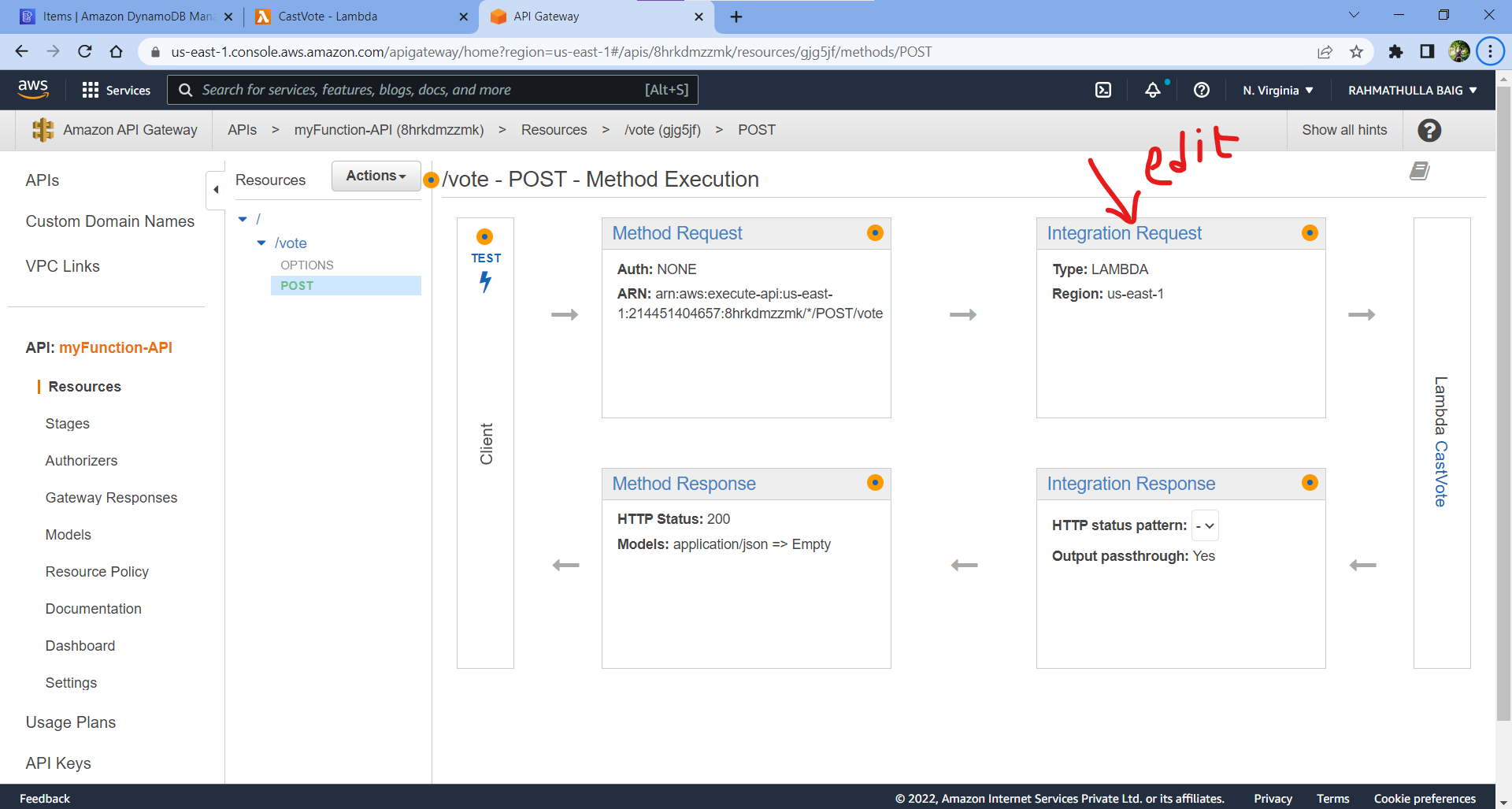
Copy the CastVote.json integrate with Dynamo DB(ADD Trigger)

Api GateWay







Create post method Add Lambda Function to it.

Add the above in content

{

"httpMethod": "POST",

"body": {

#foreach( $token in $input.path('$').split('&') )

#set( $keyVal = $token.split('=') )

#set( $keyValSize = $keyVal.size() )

#if( $keyValSize >= 1 )

#set( $key = $util.urlDecode($keyVal[0]) )

#if( $keyValSize >= 2 )

#set( $val = $util.urlDecode($keyVal[1]) )

#else

#set( $val = '' )

#end

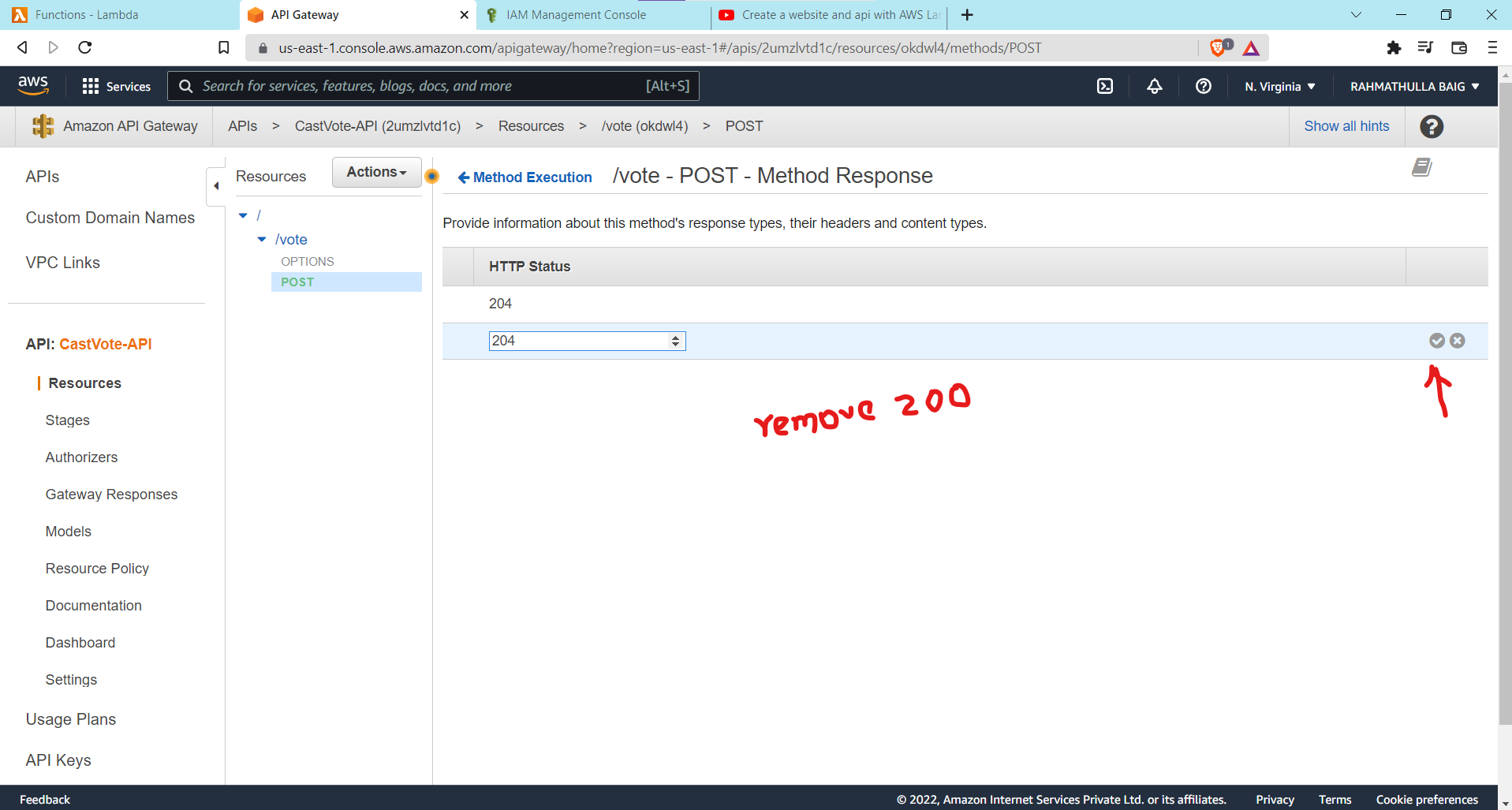
"$key": "$val"#if($foreach.hasNext),#end

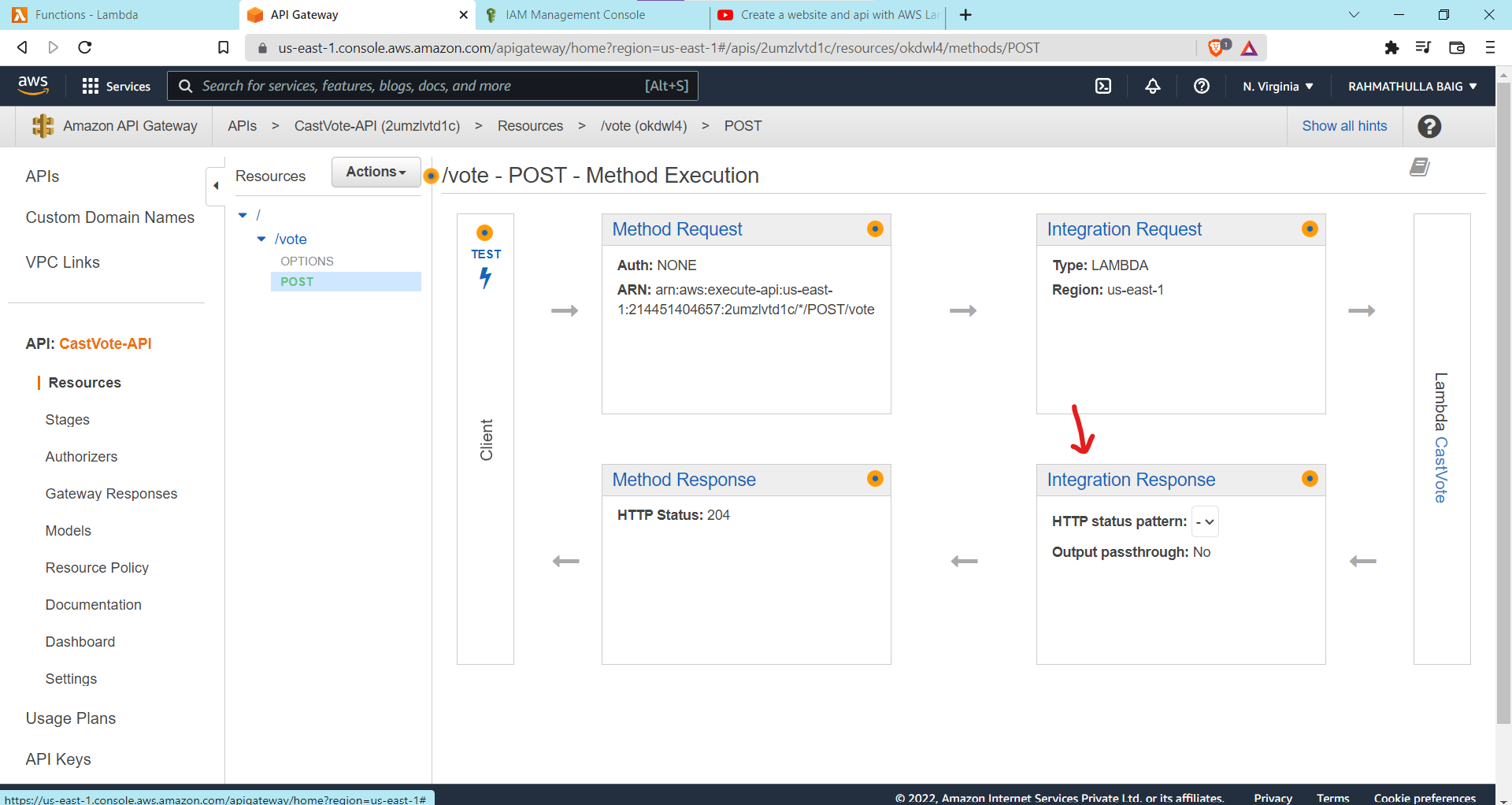
#end

#end

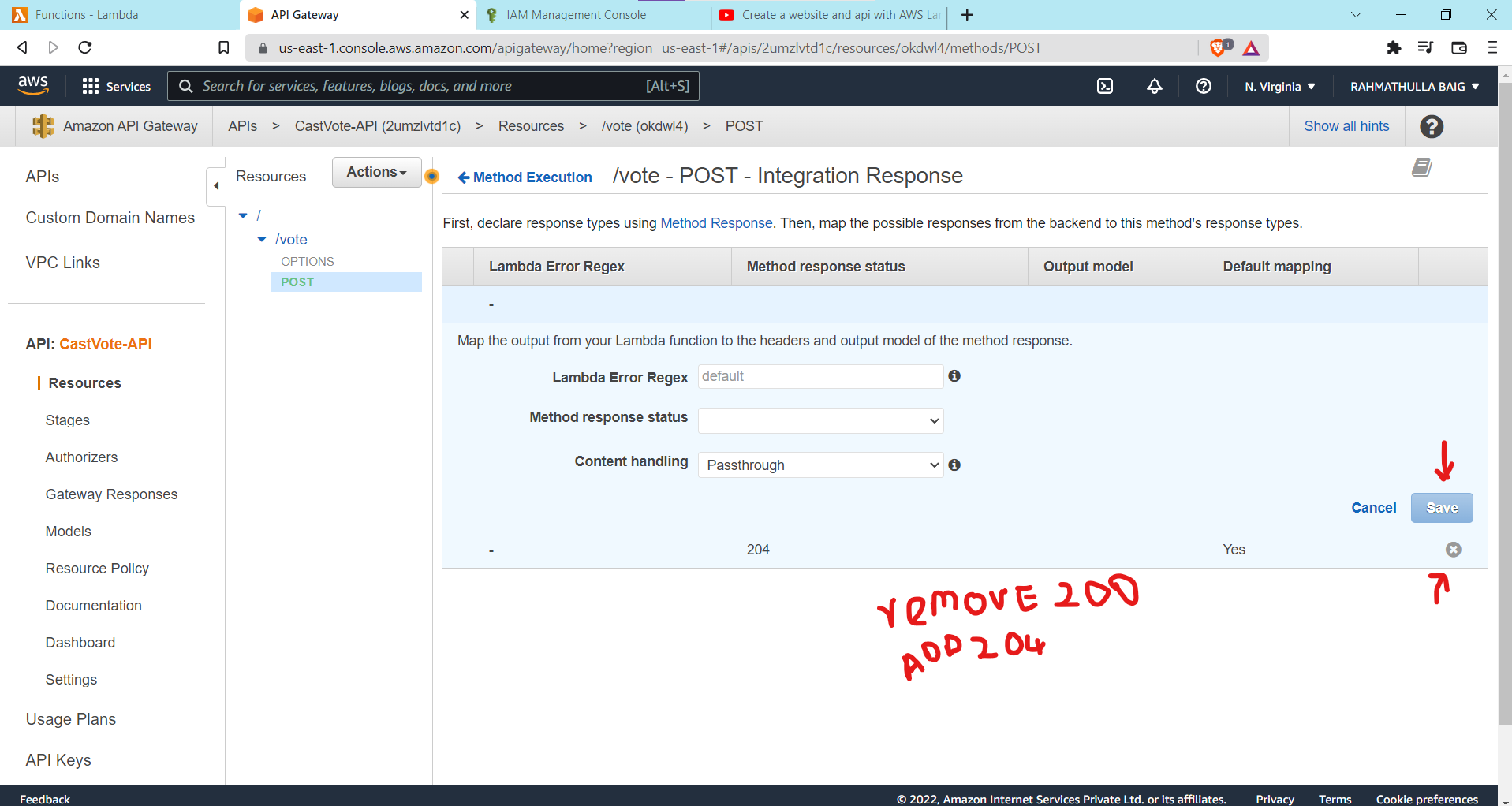
}

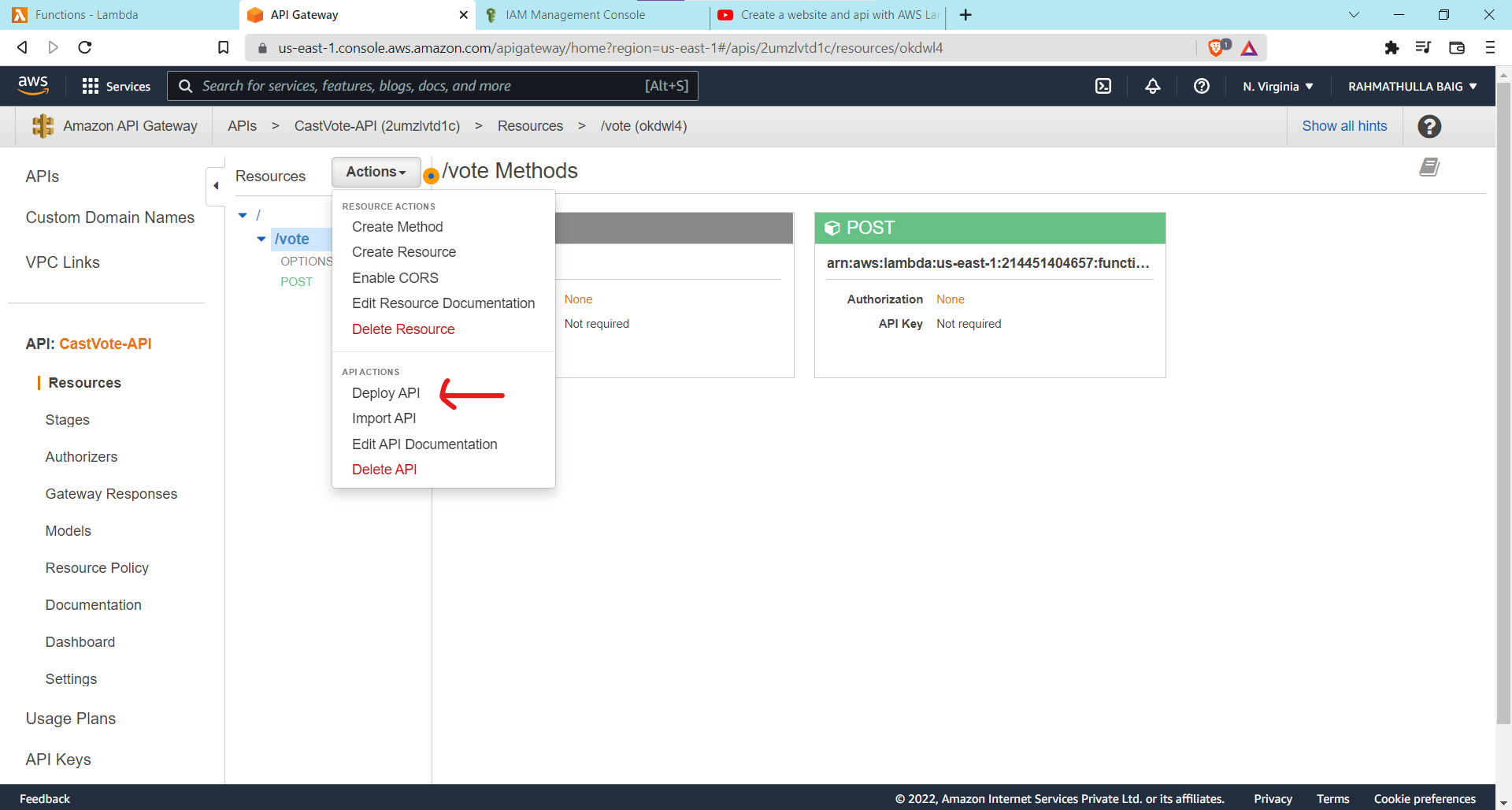
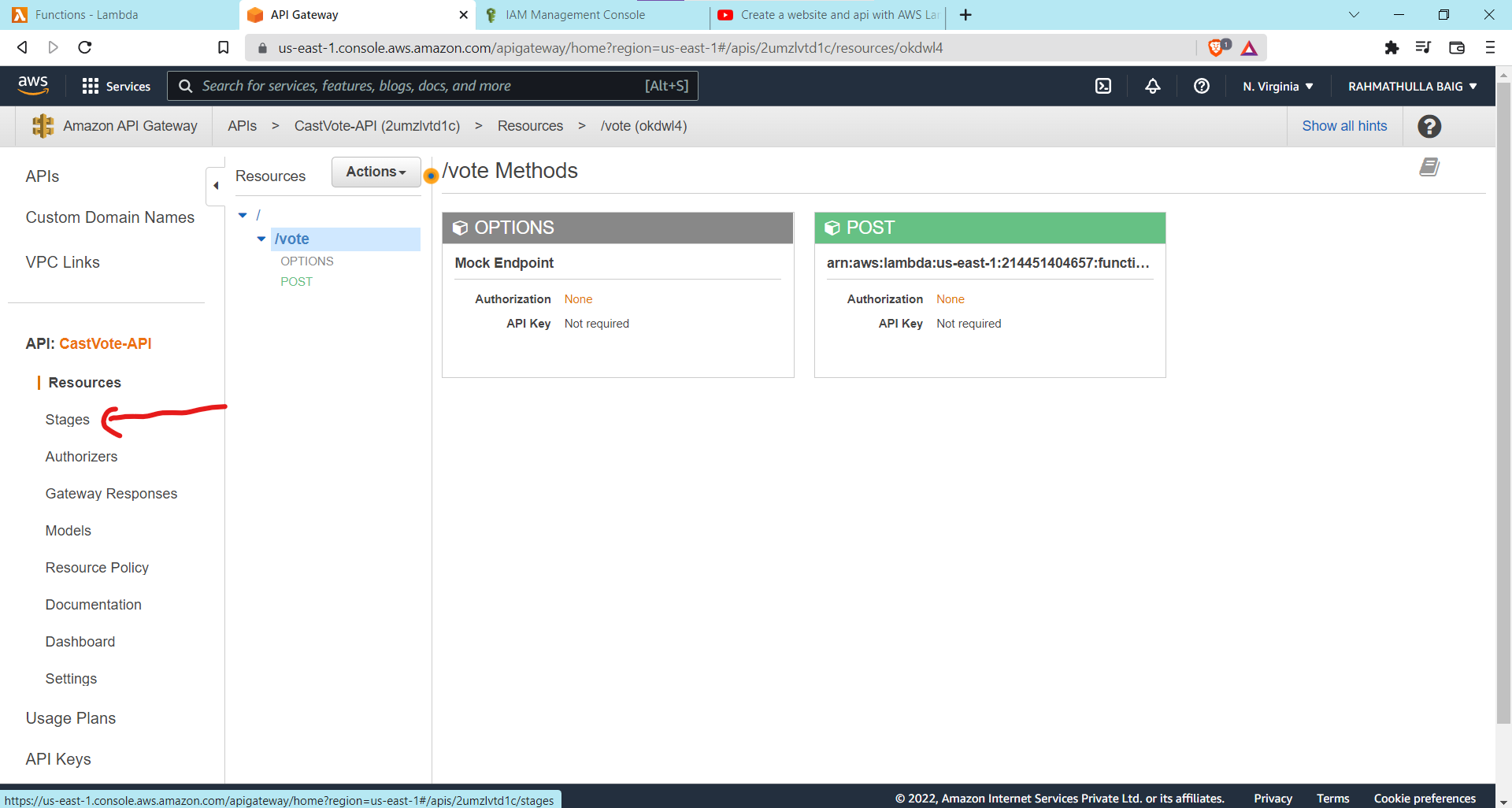
}

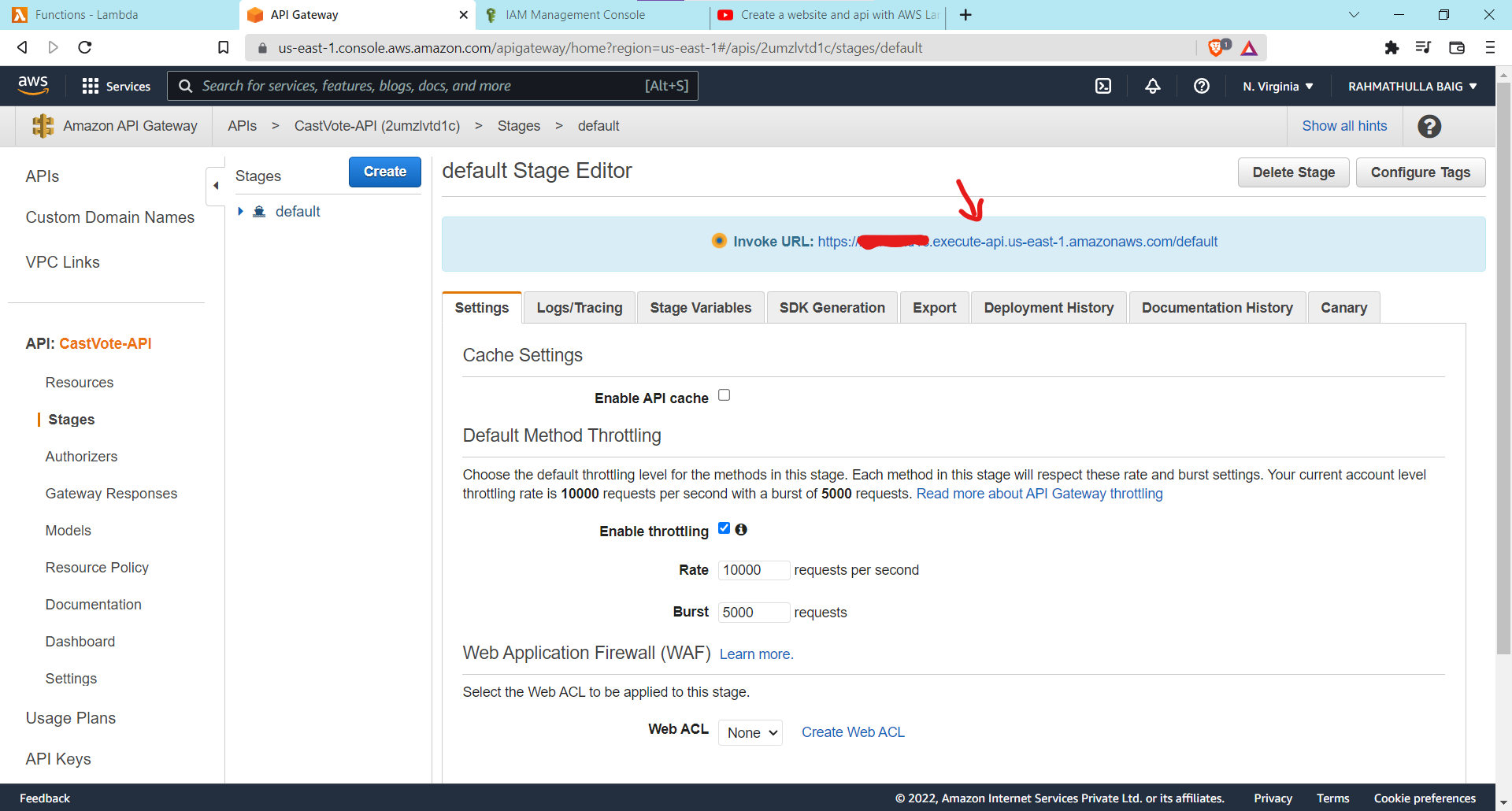




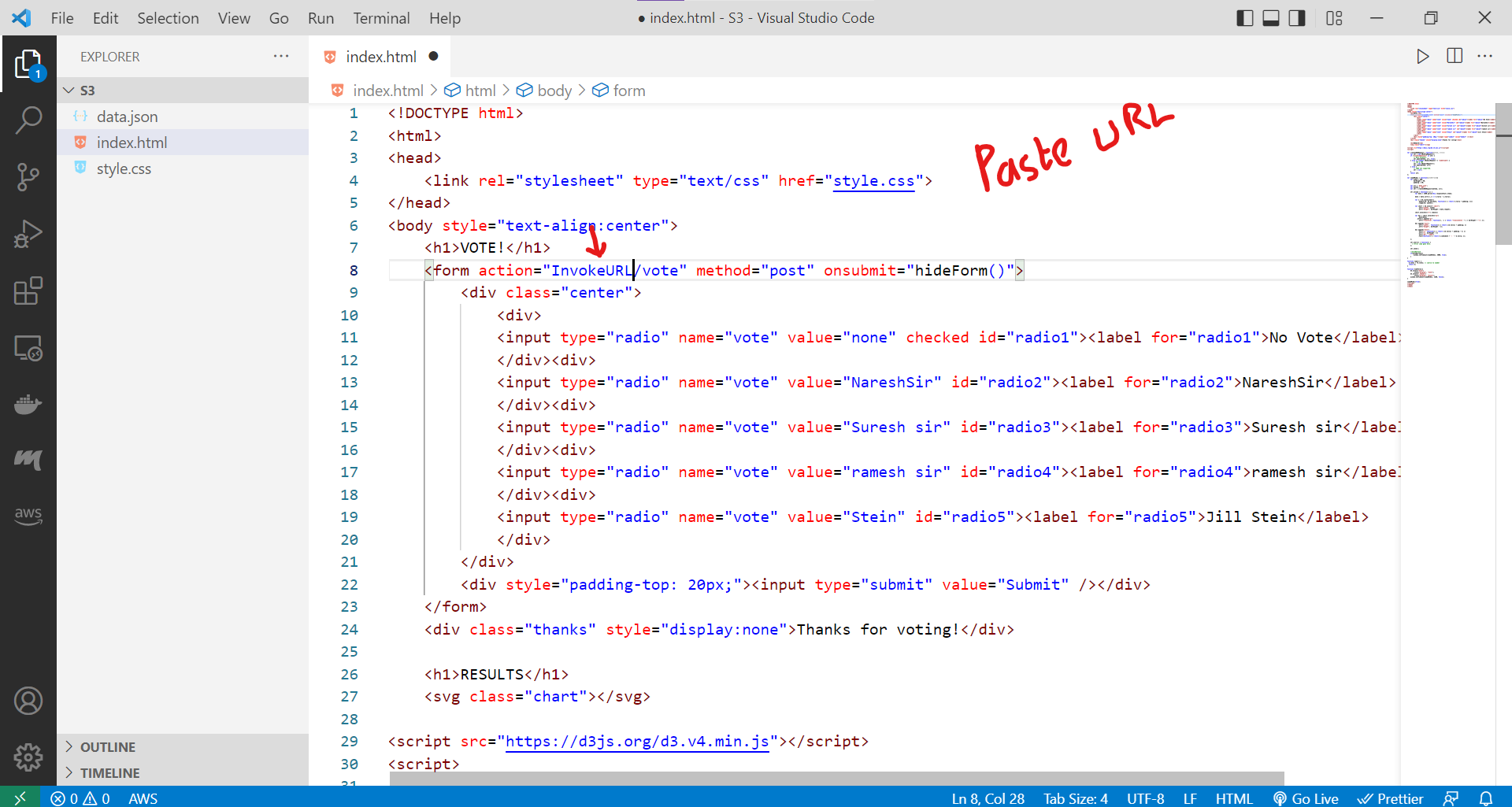
In Integration response Add 204 in default and remove 200.

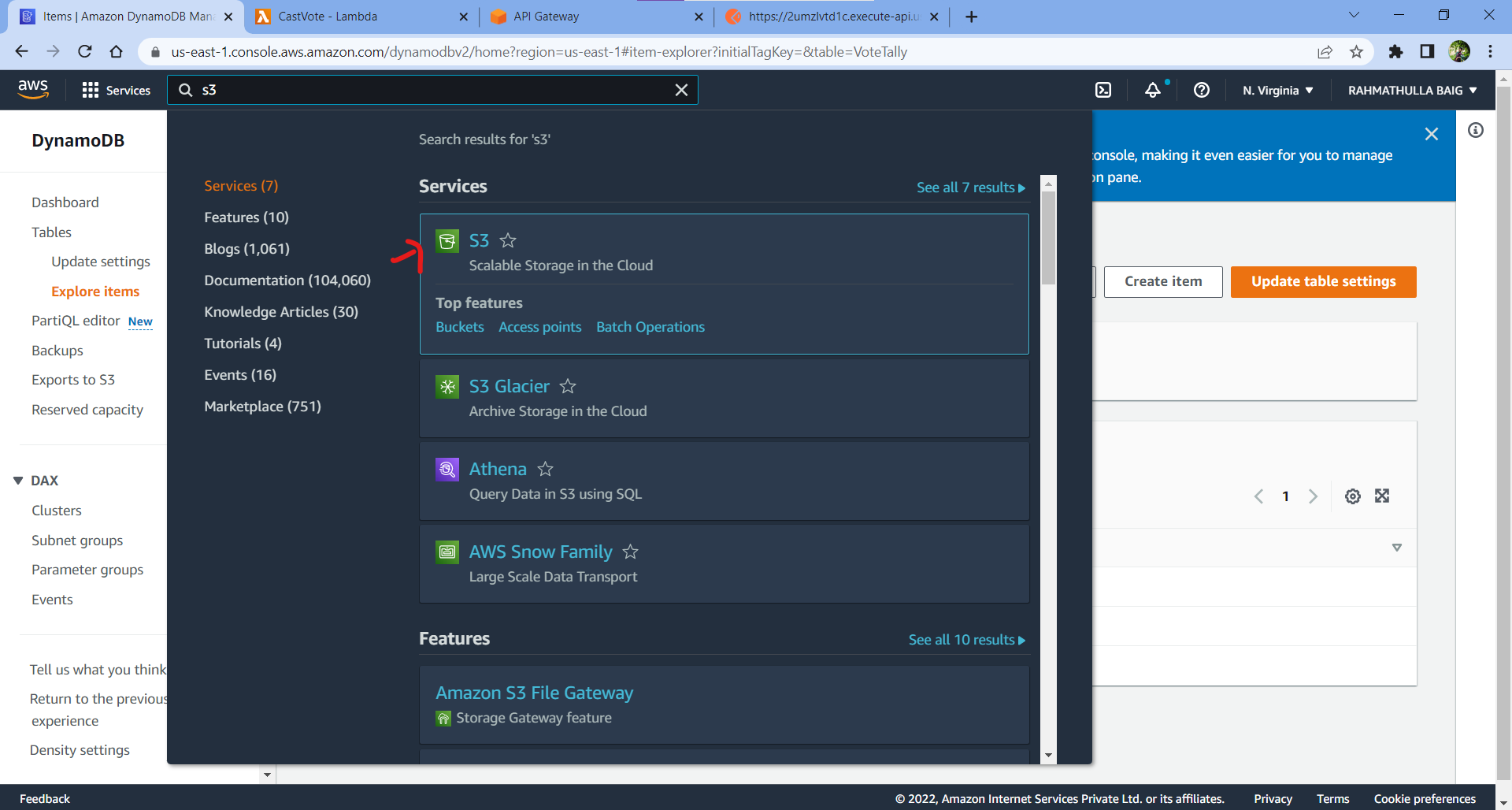


Deploy API  

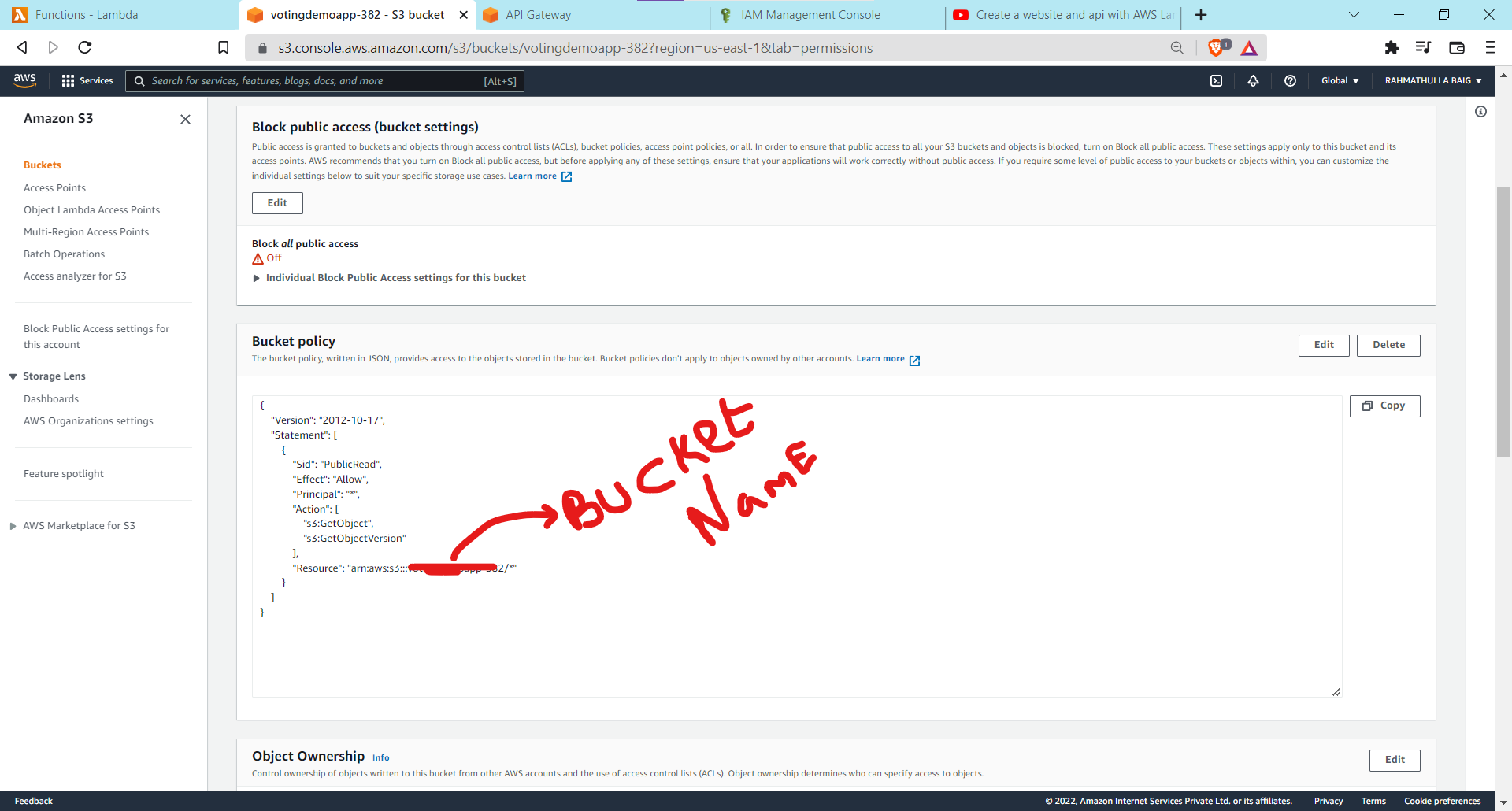


Copy Invoke URL and add it in index.html





Add s3 files in bucket with default settings add bucket policy and enable static Website Hosting



Bucket Policy

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicRead",

"Effect": "Allow",

"Principal": "\*",

"Action": [

"s3:GetObject",

"s3:GetObjectVersion"

],

"Resource": "arn:aws:s3:::YourBucketName/\*"

}

]

}

Go to S3 Bucket Properties -> static hosting ->Copy Url

