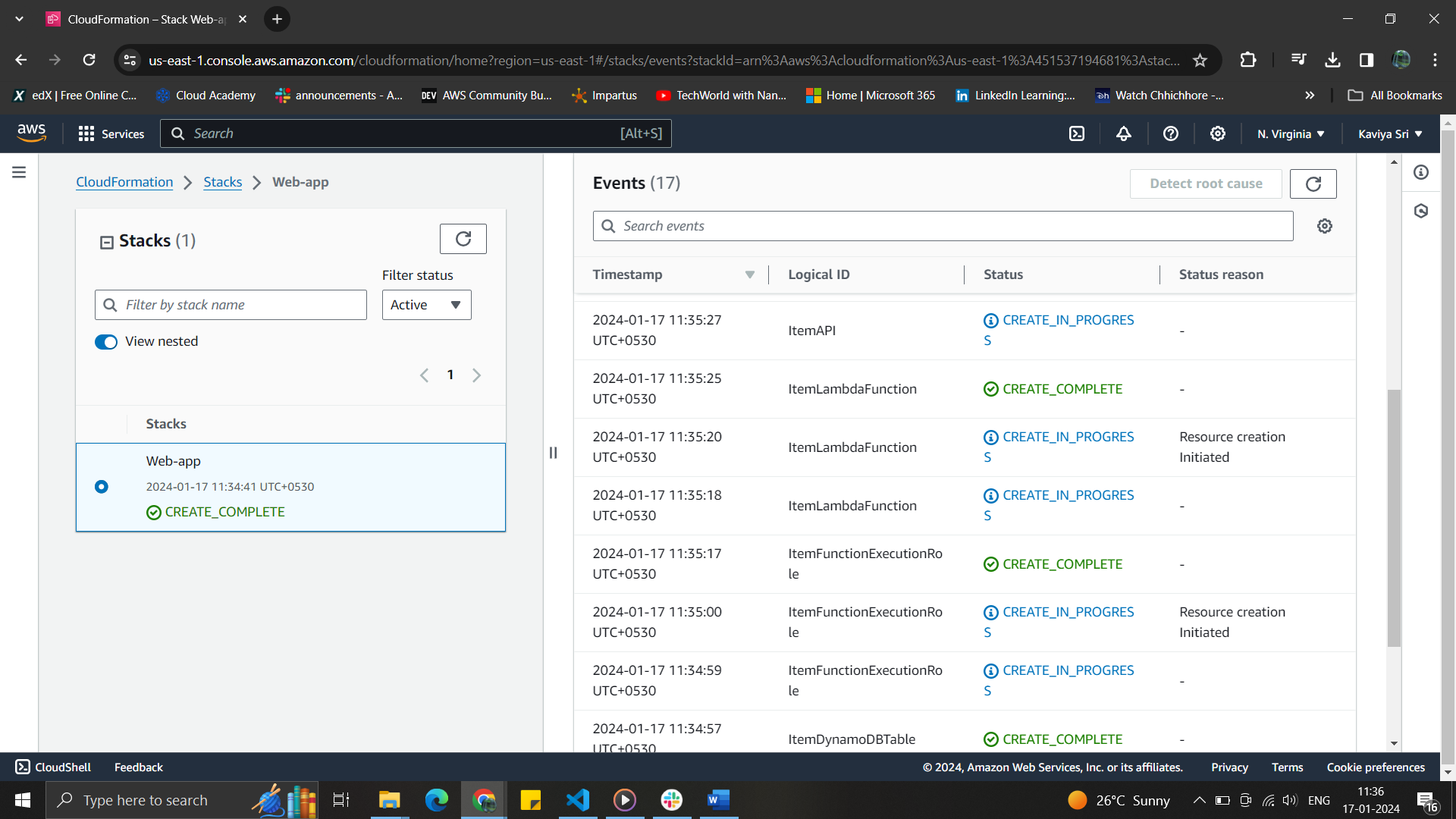
Serverless Web Application – Task 2

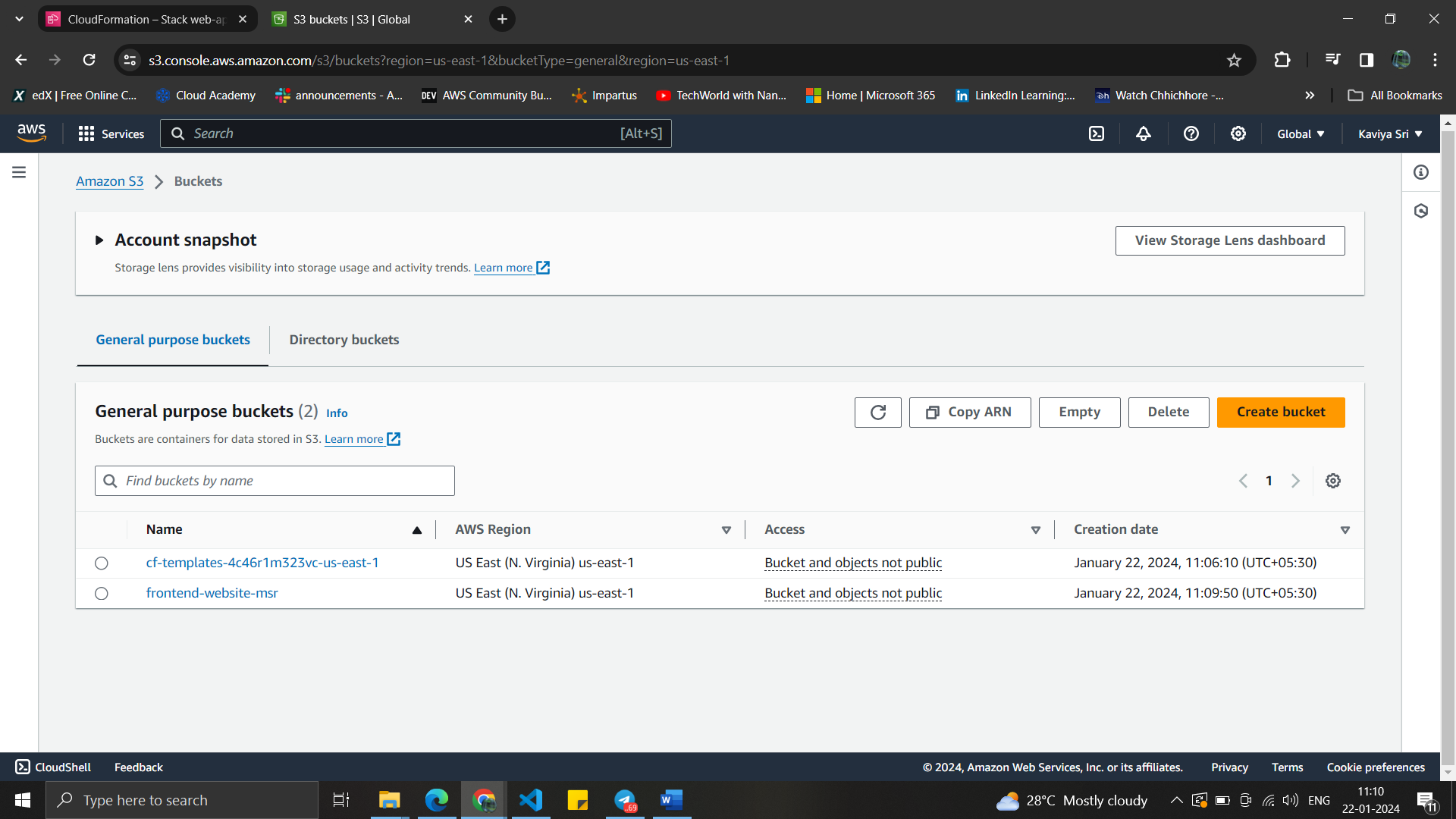
* Kaviya Sri A N

Created the cloud formation stacks for DynamoDB , Lambda and API Gateway

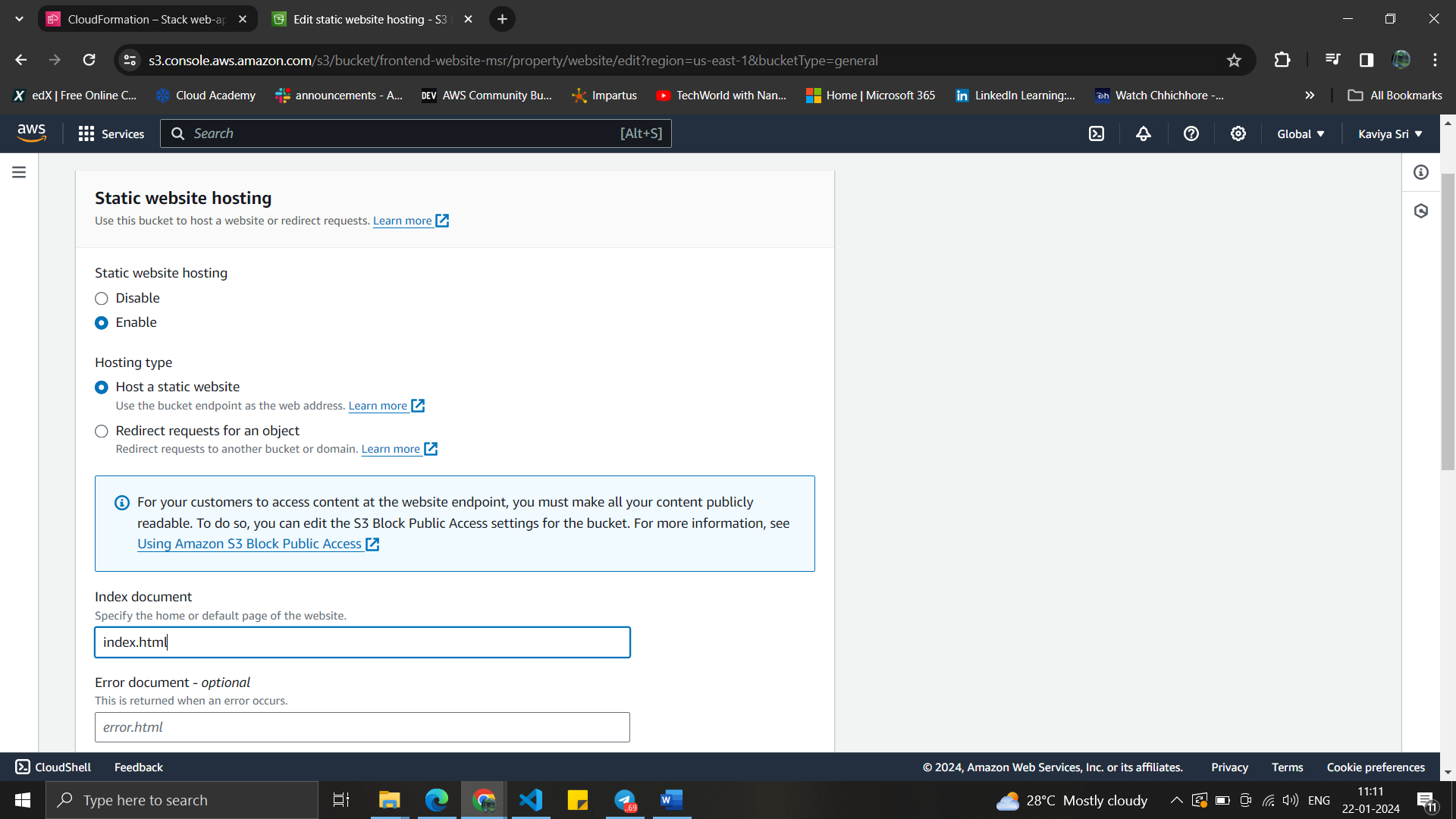




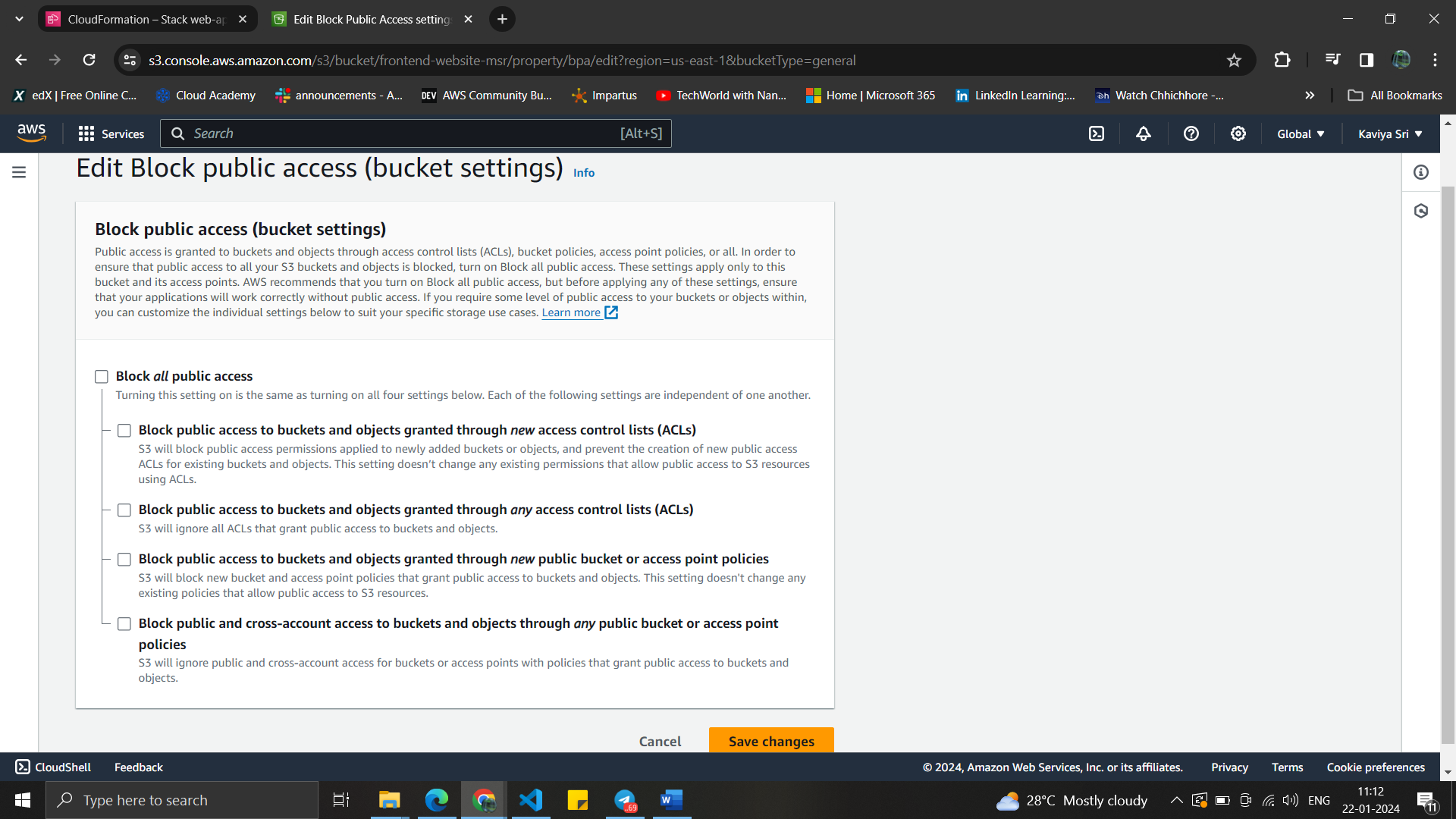
Created the bucket for the website



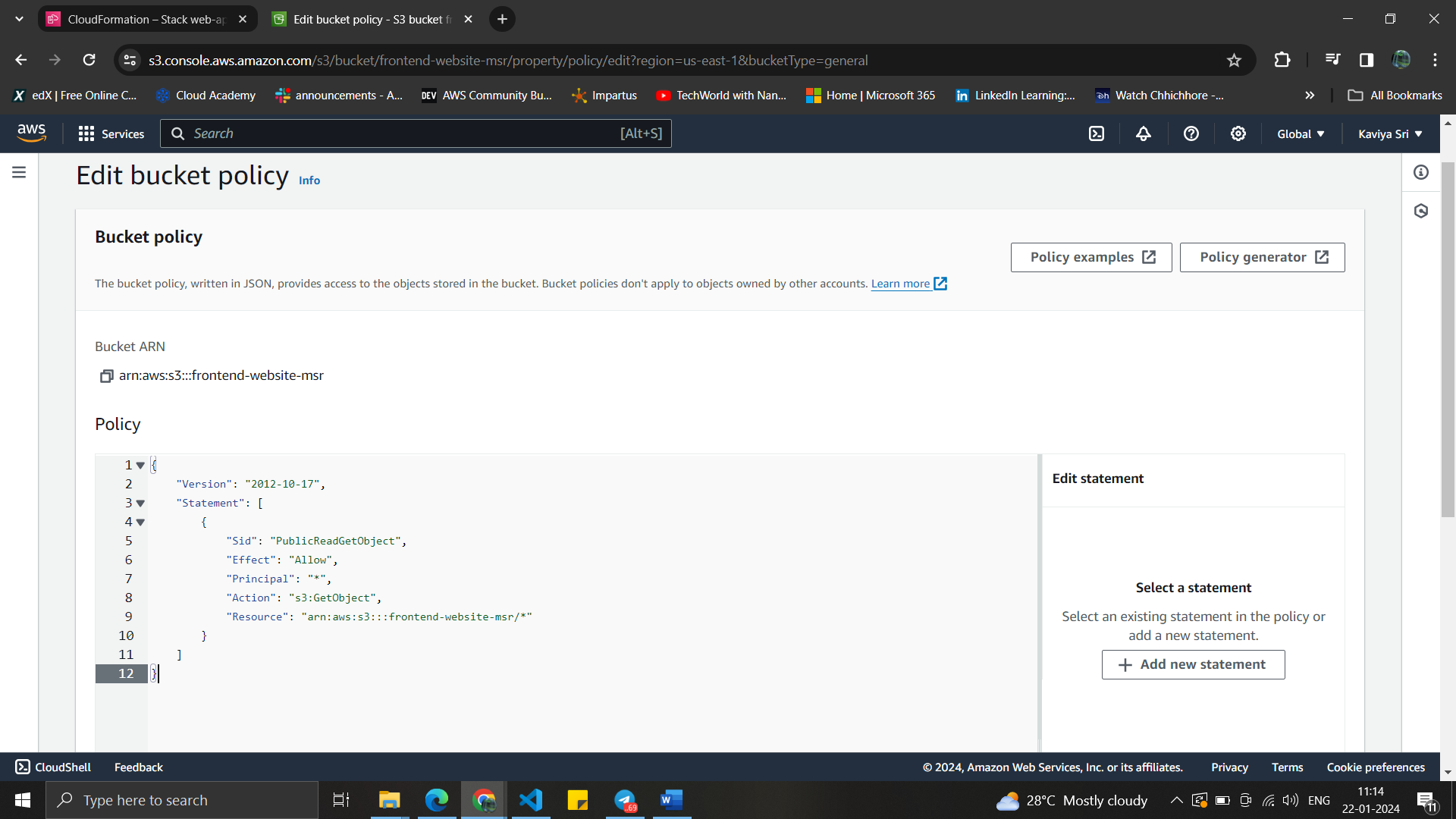
Enabled the static website hosting



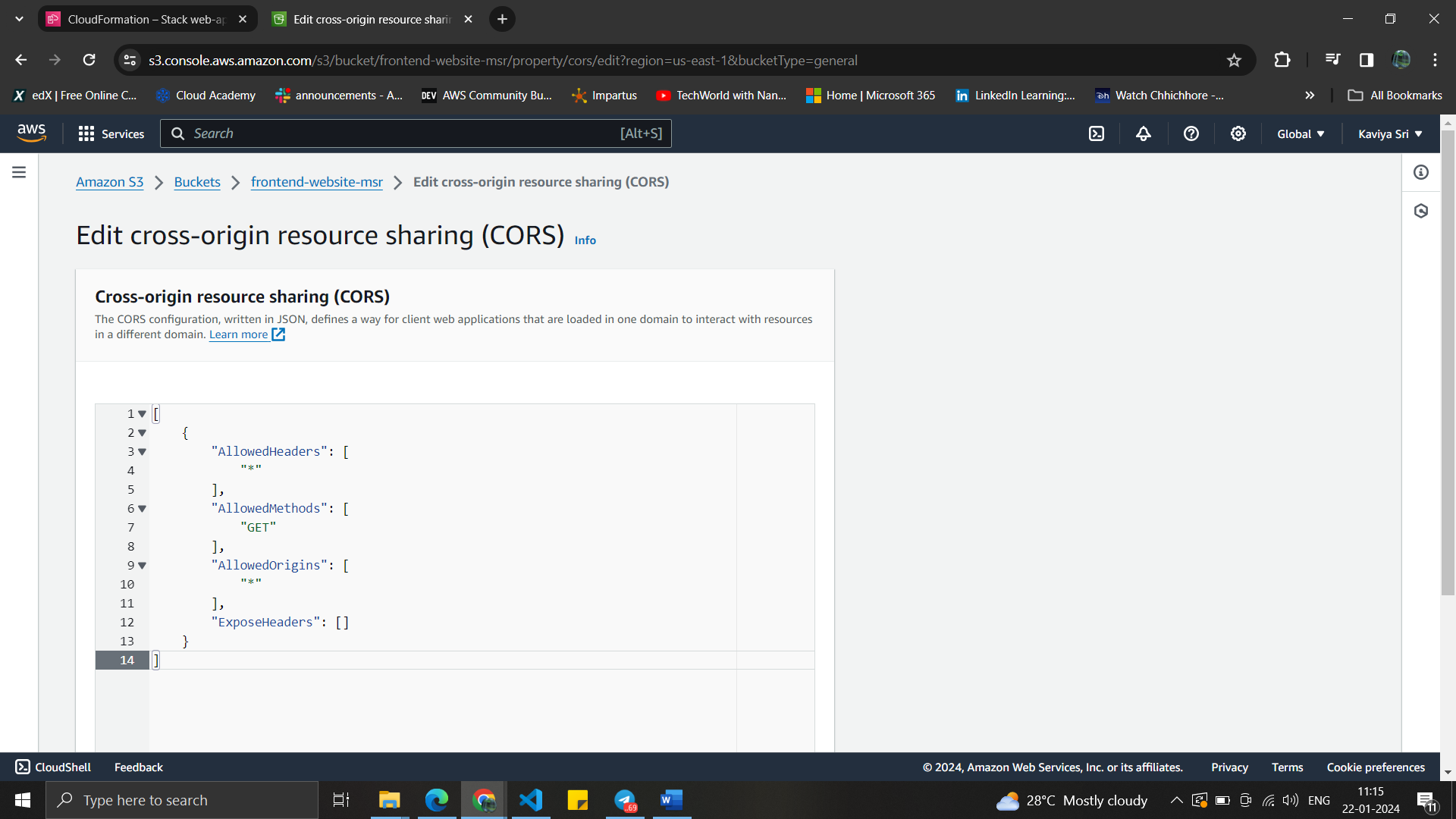
Made the bucket as public access



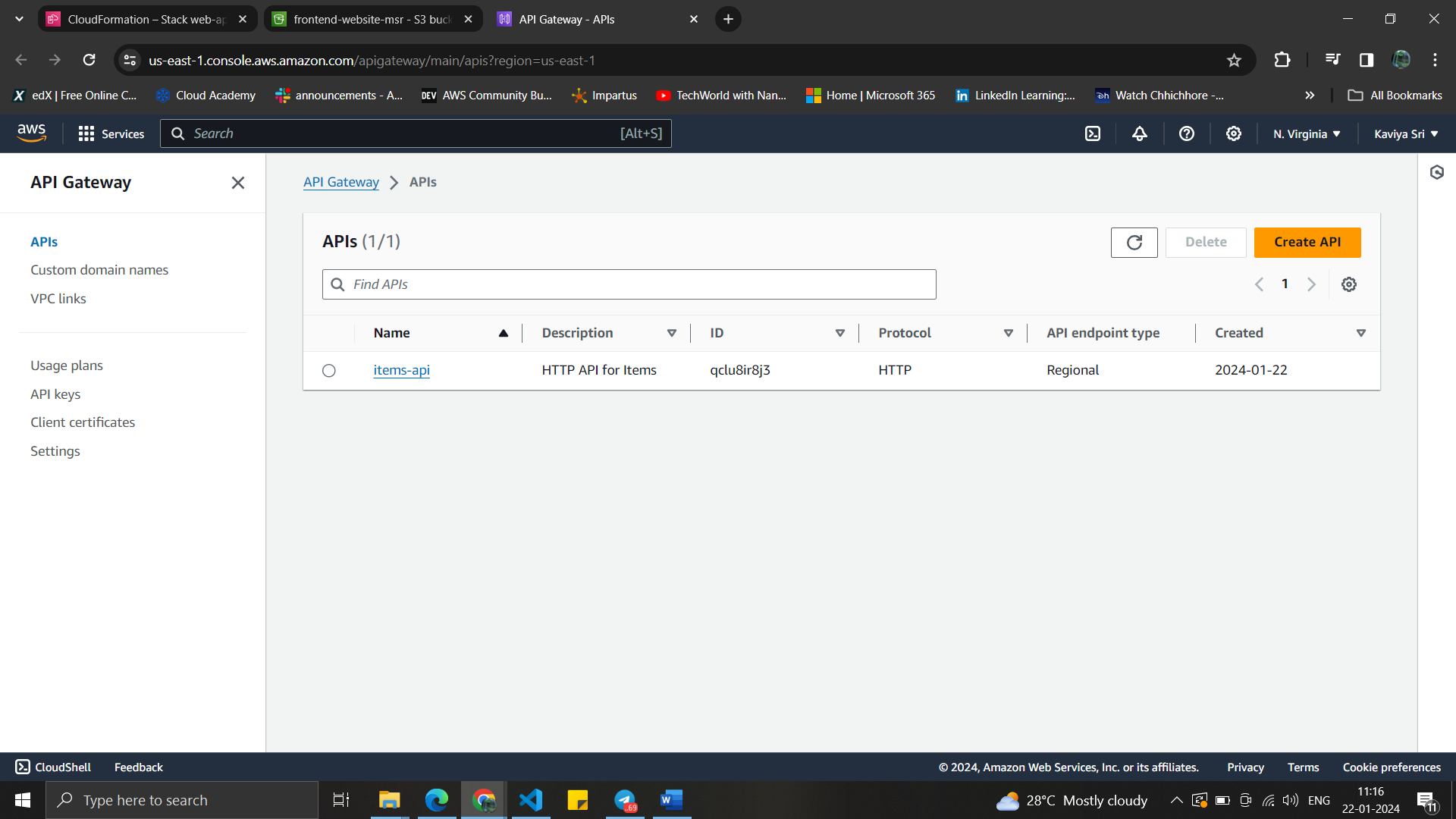
Bucket policy is configured with bucket ARN



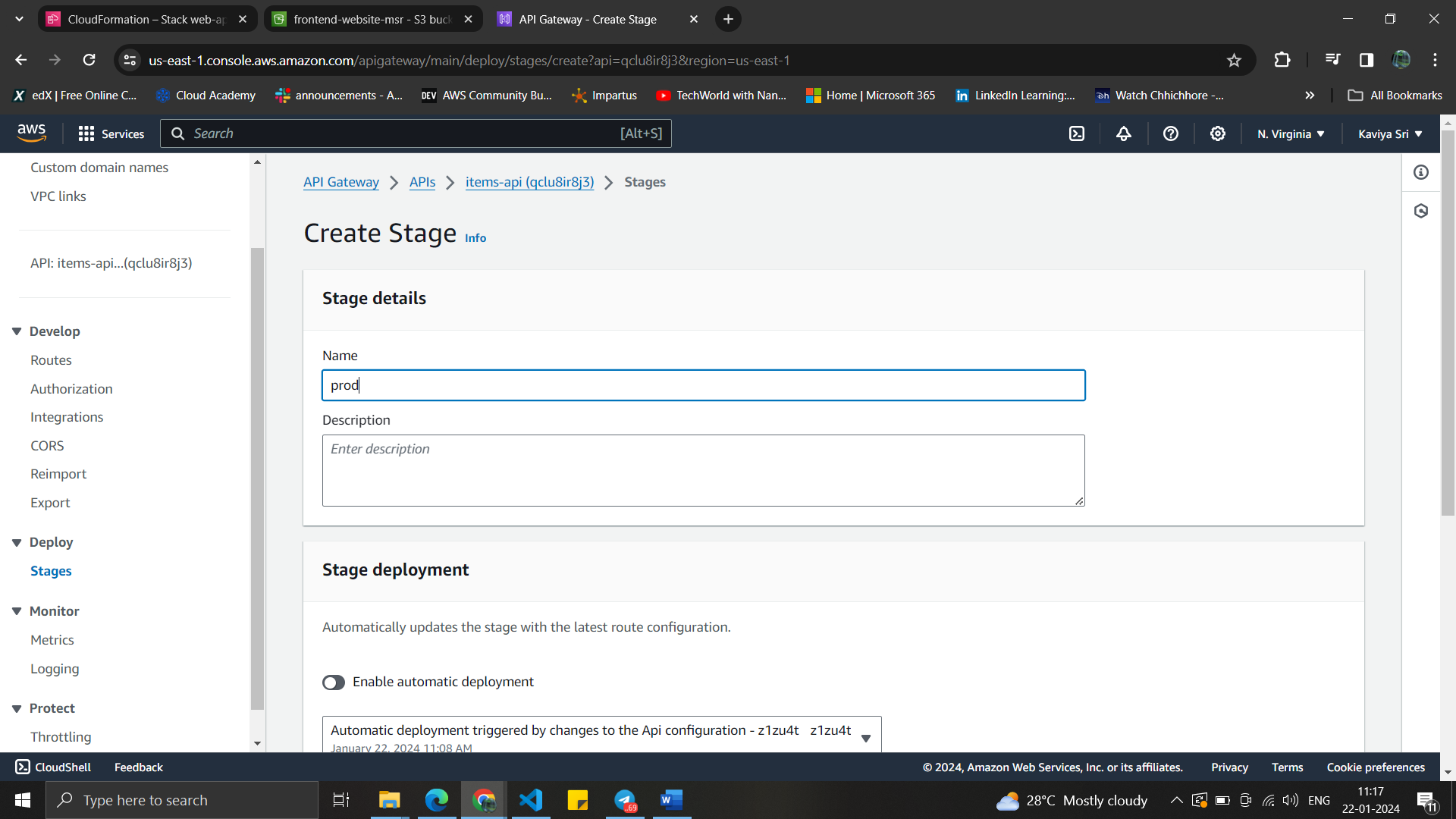
Cross-origin resource sharing (CORS) is configured



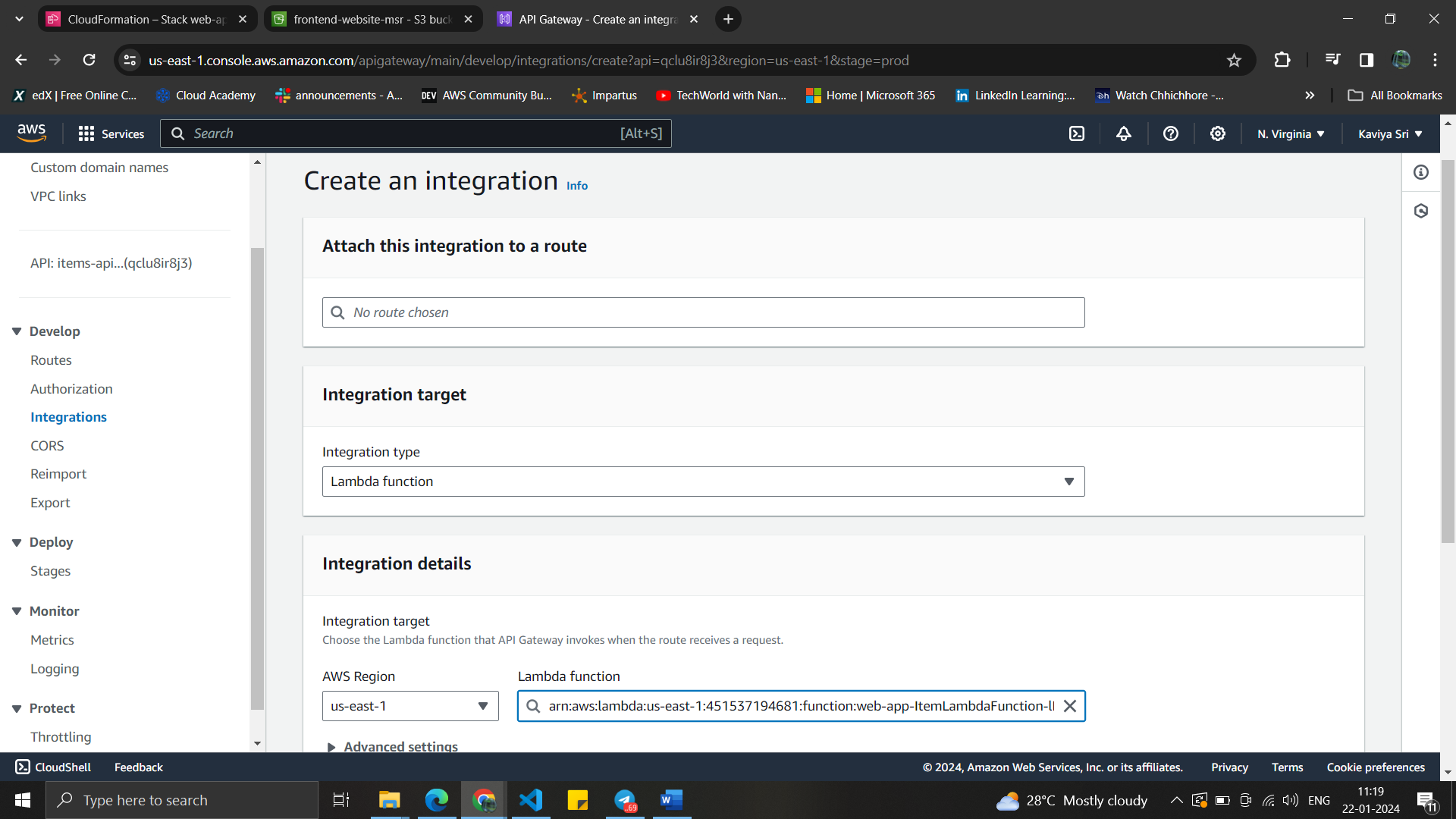
API Gateway console with already created stack “items-api”



Stage section is configured .

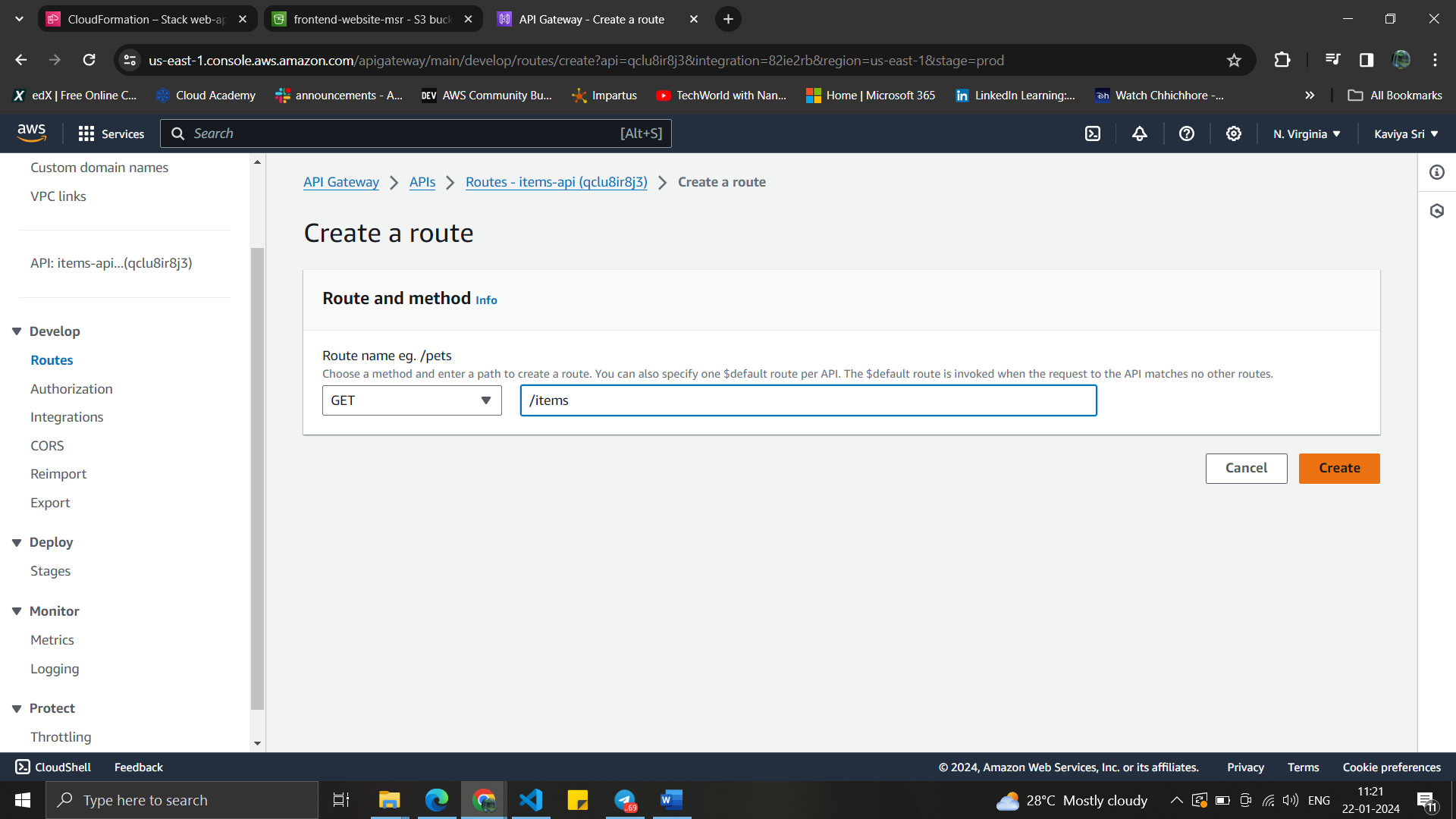


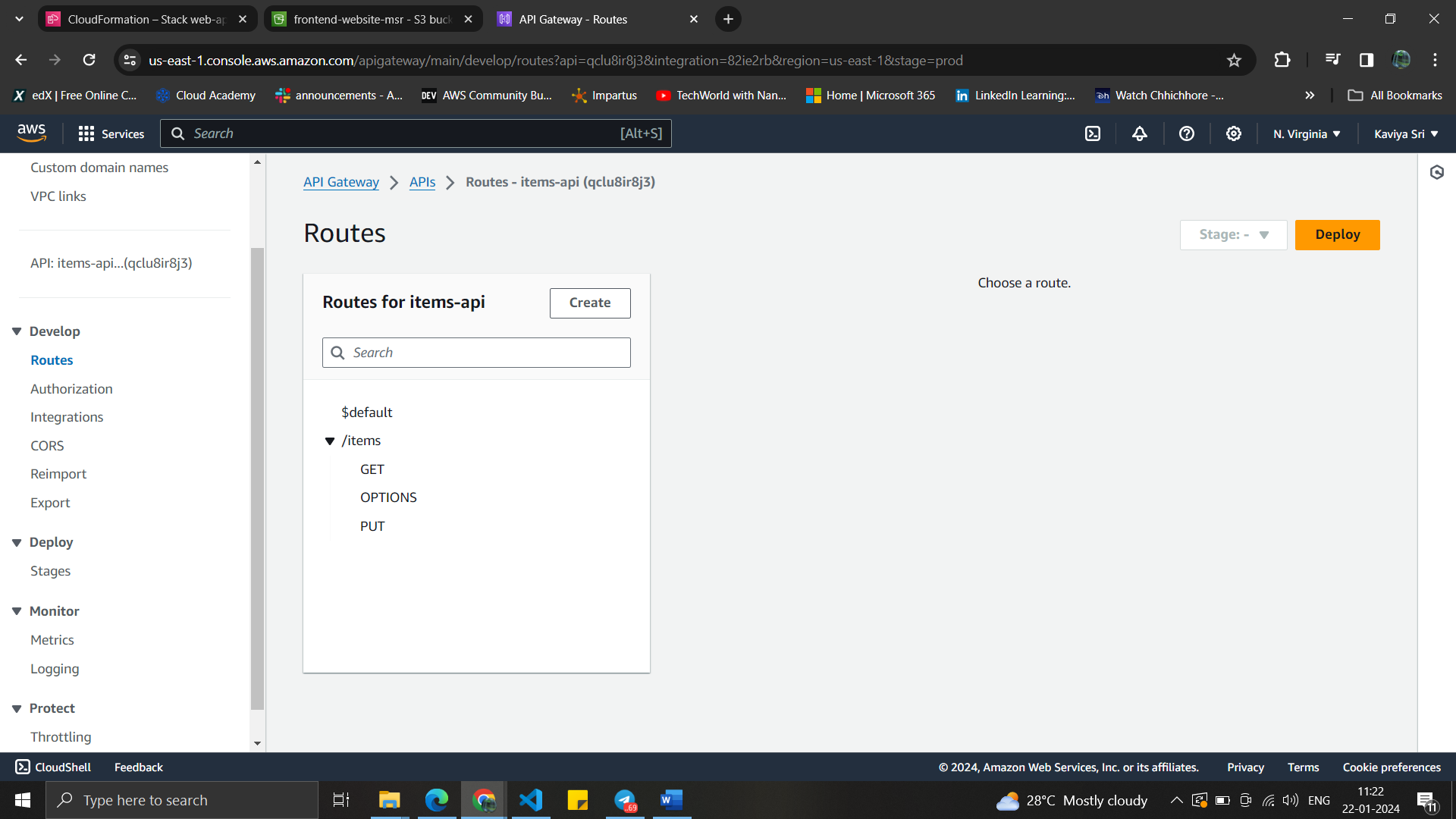
Integration section is configured with Lambda function as target.



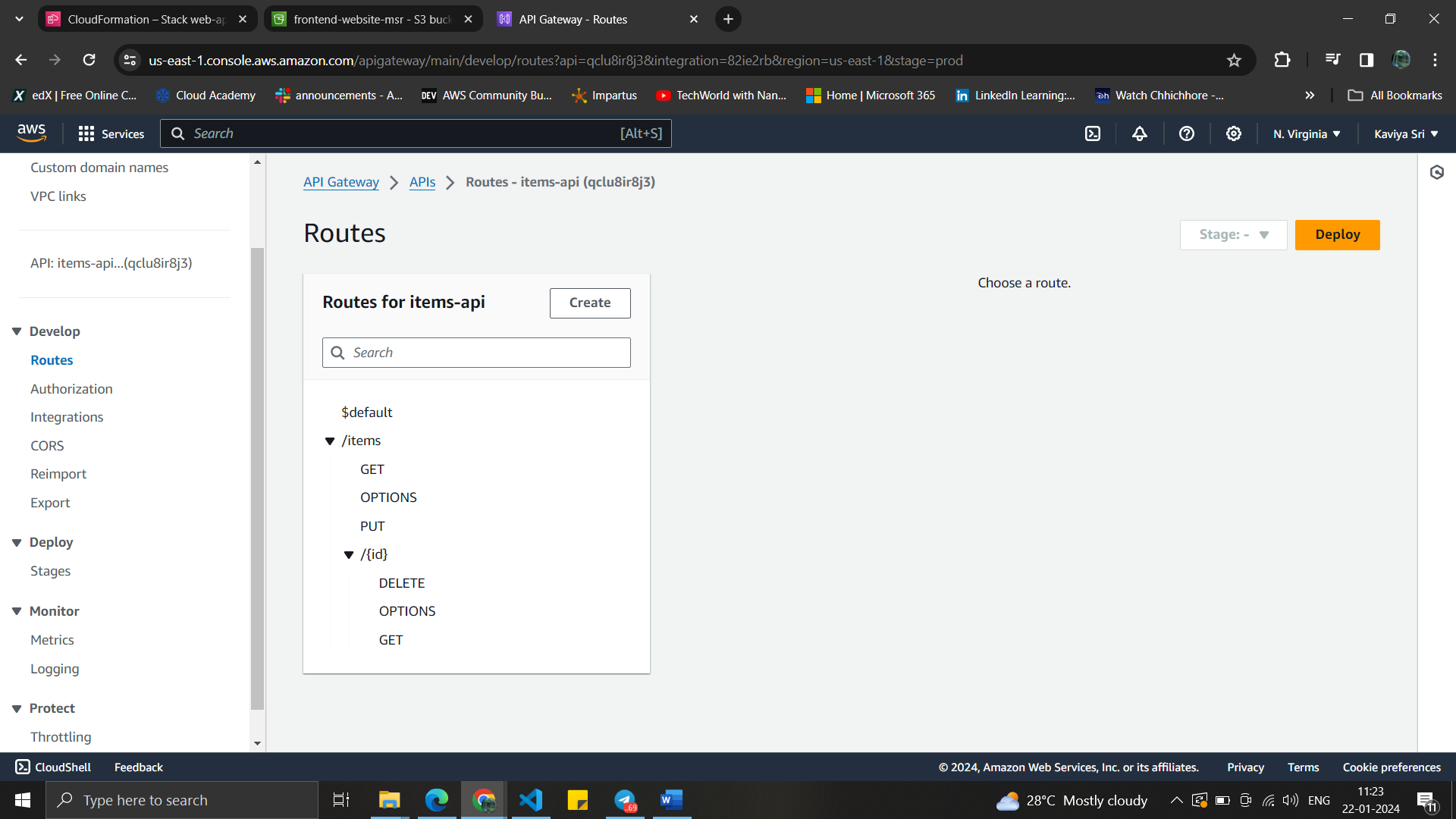
Integration Id : 82ie2rb

Route section is configured by creating the routes for put, get ,delete , options

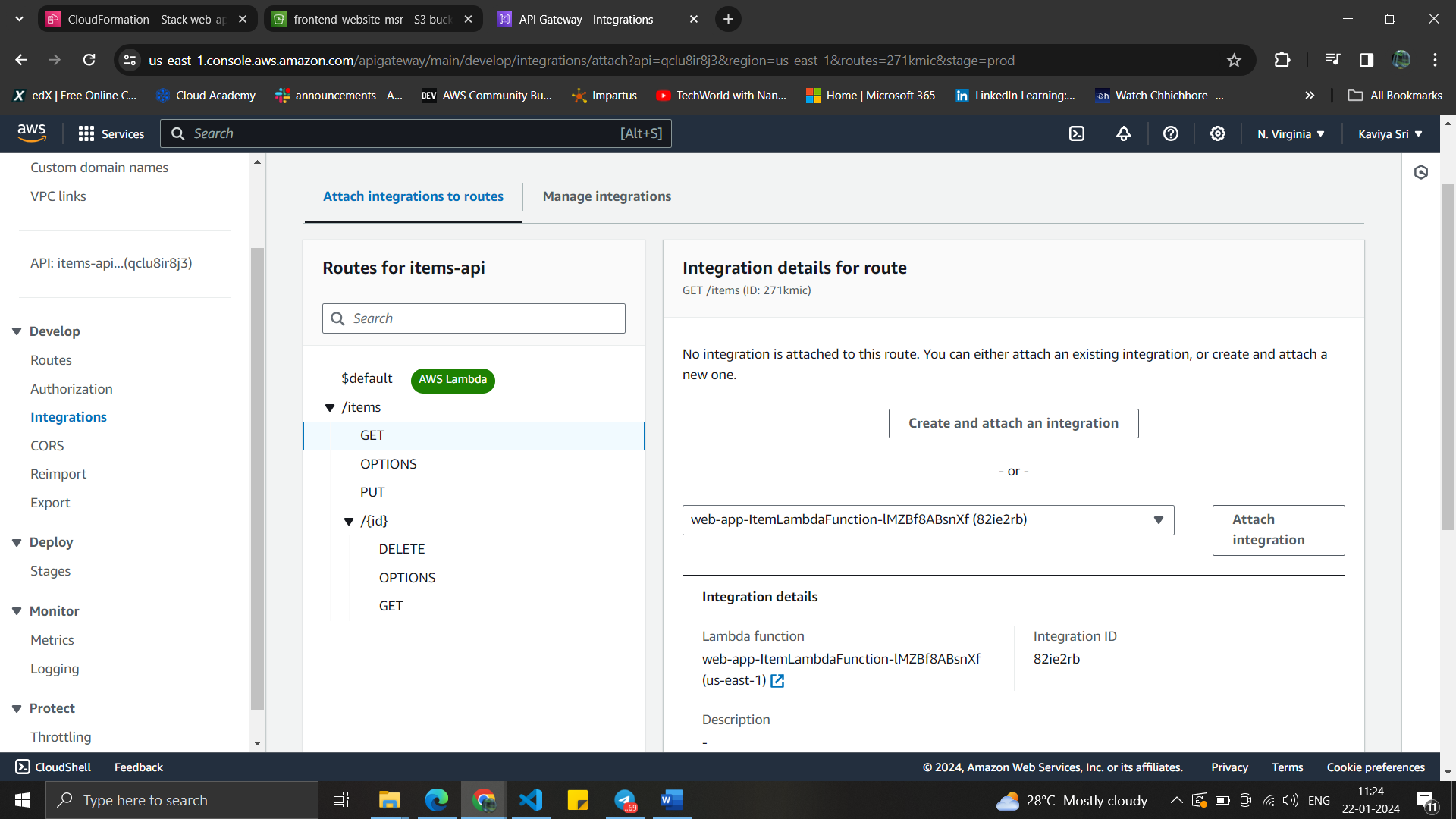




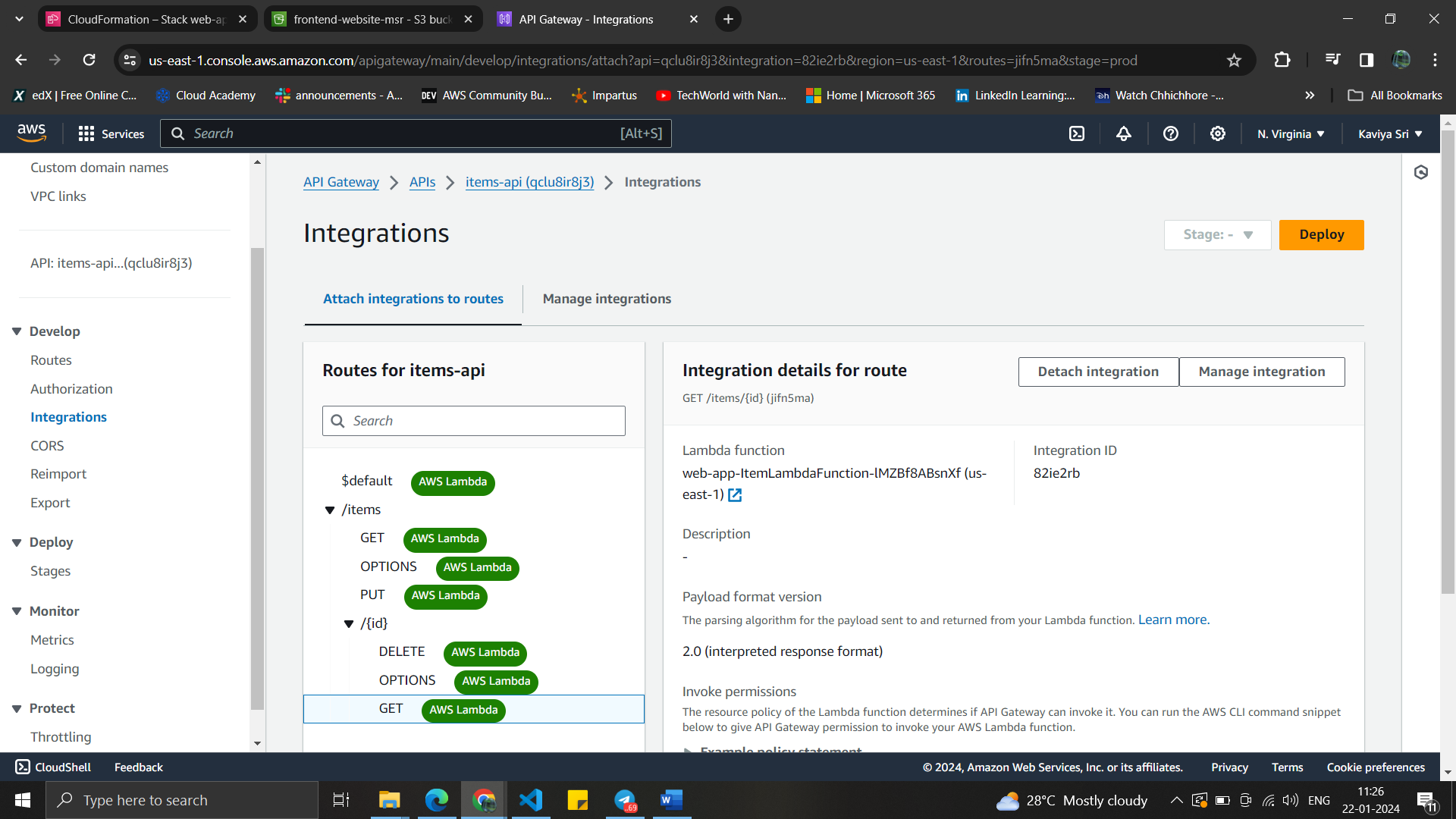
Below is the all the routes that are created.



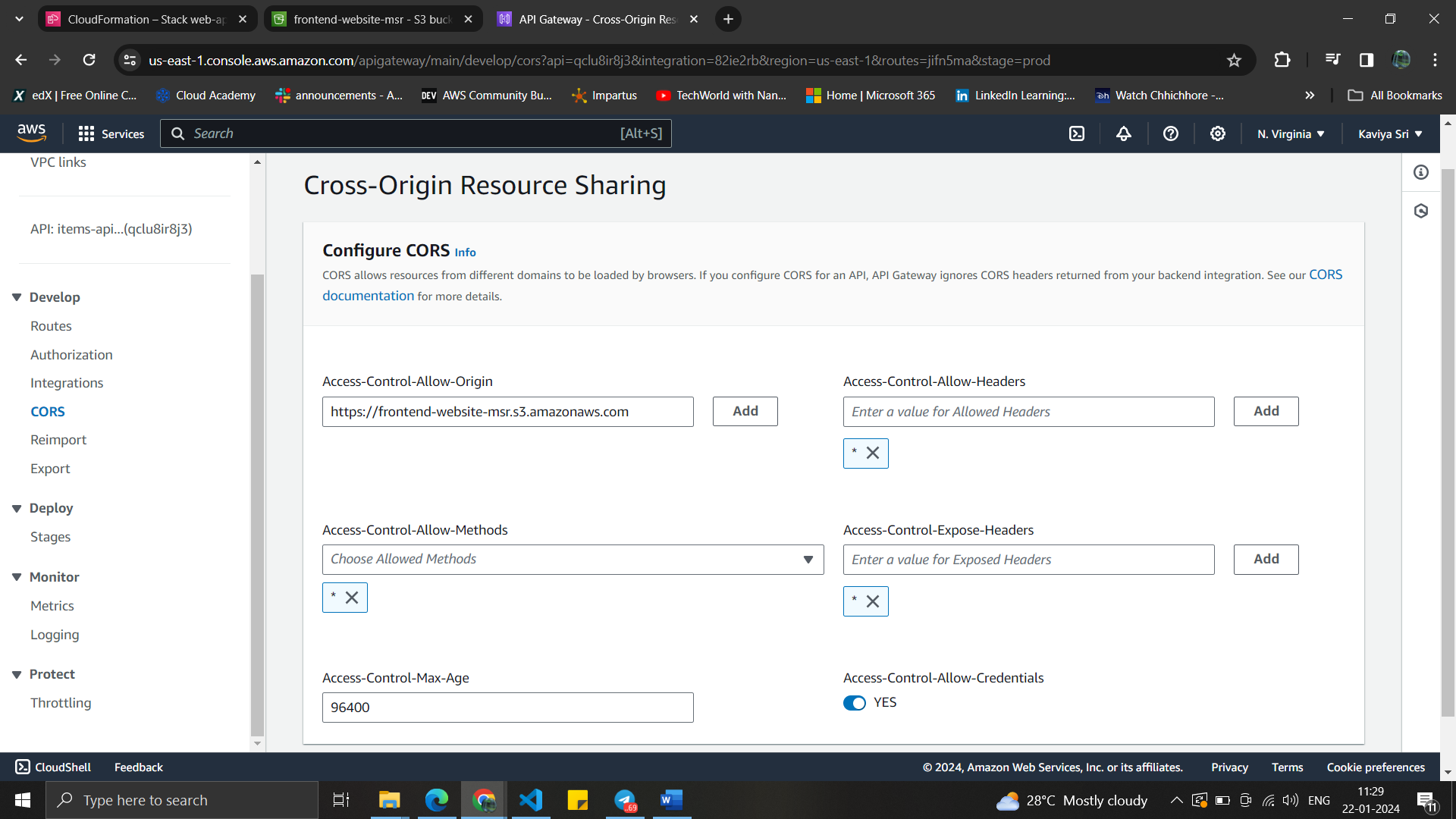
Each route is integrated with the integration-id .

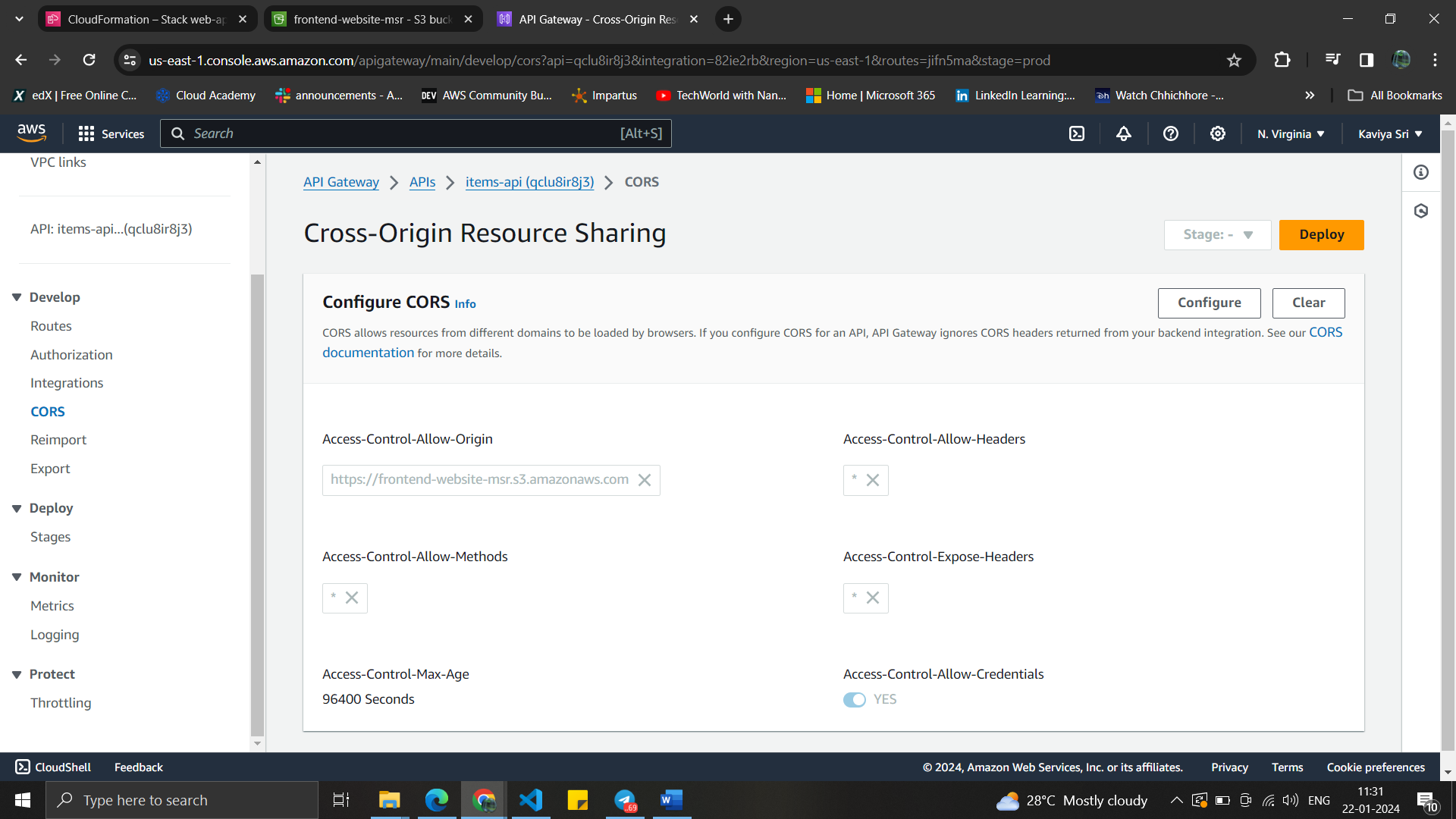


All routes are integrated with the AWS Lambda

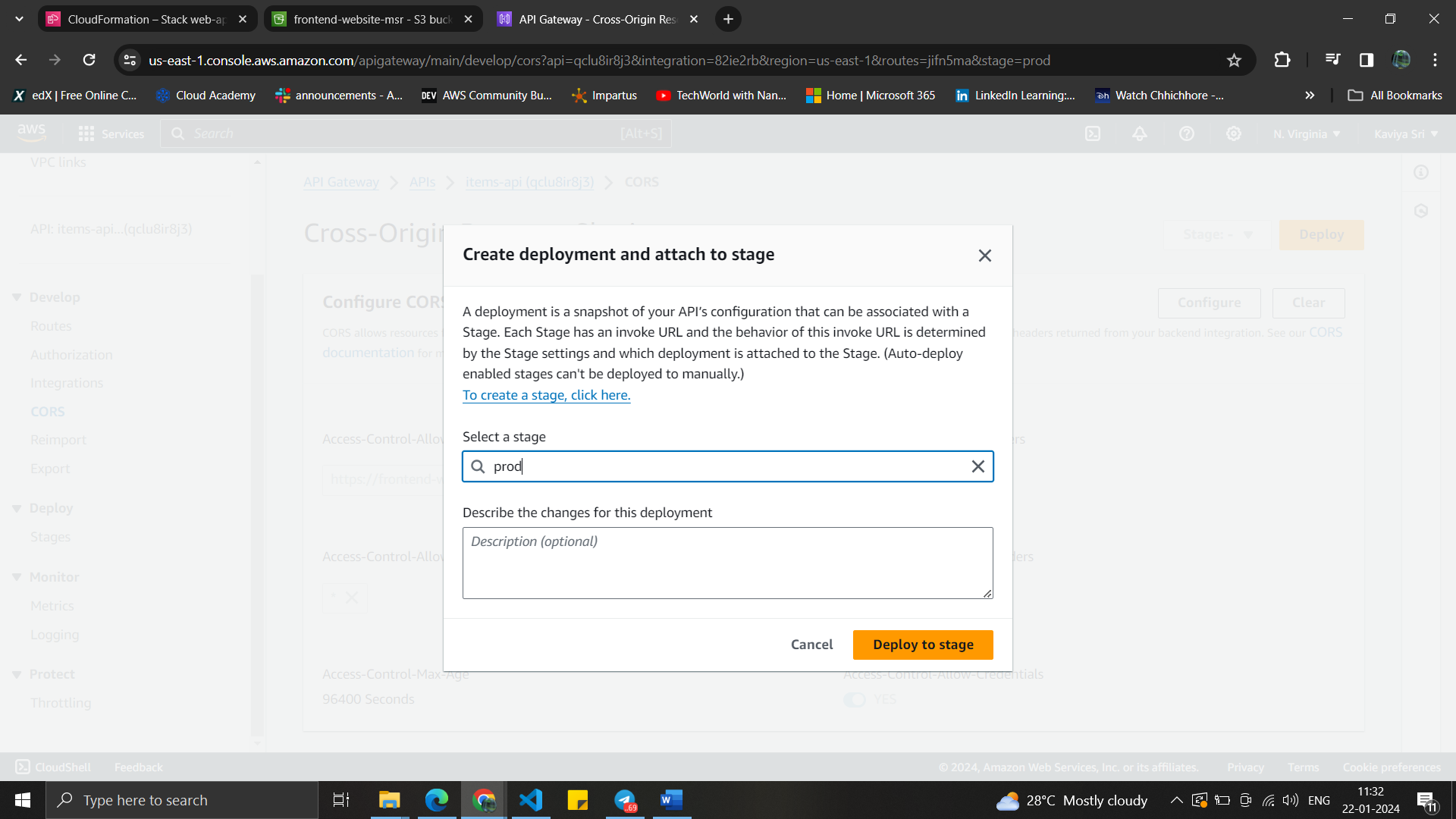


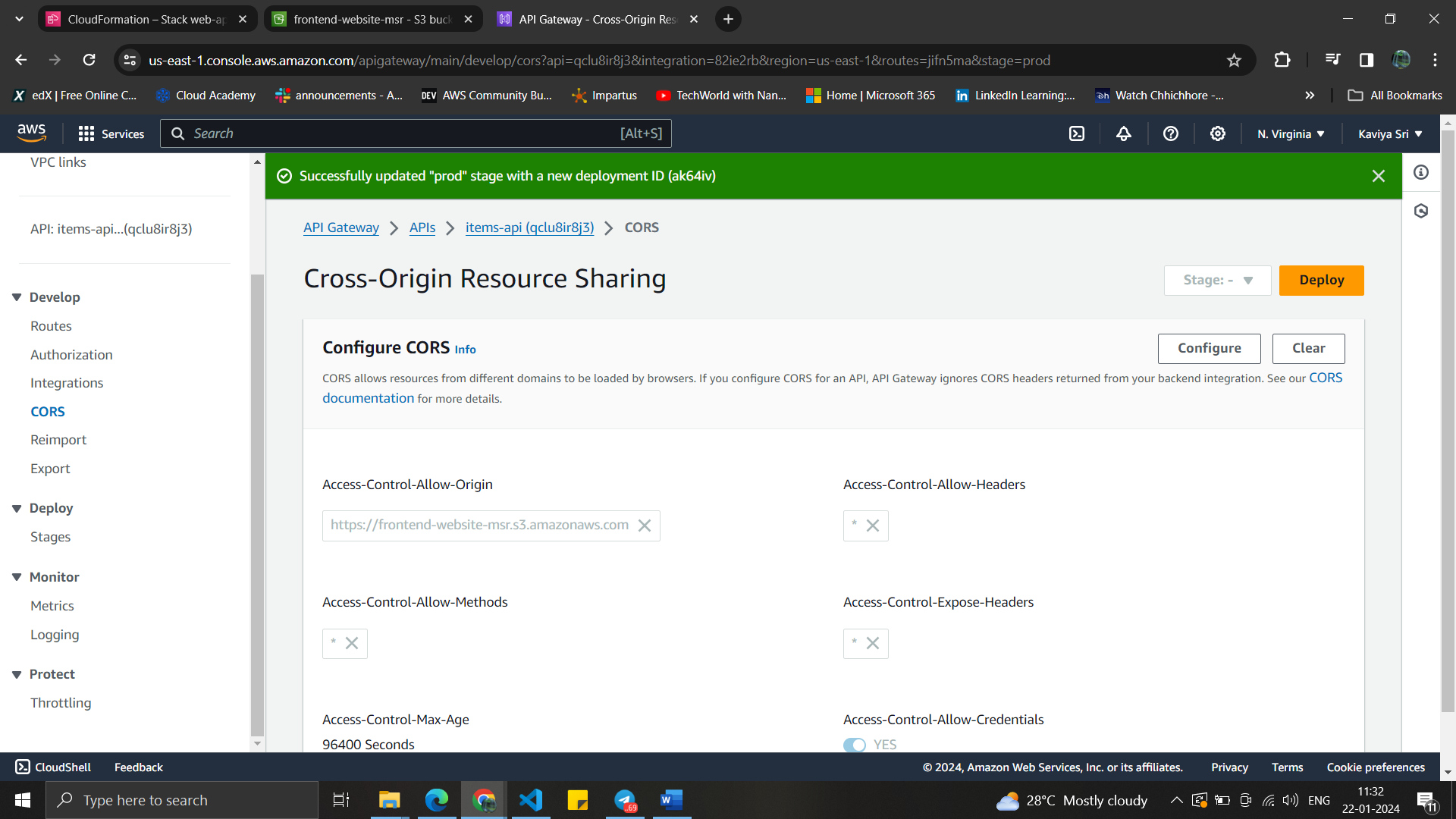
CORS section is configured with Access-Control-Allow-Origin for the specific bucket.



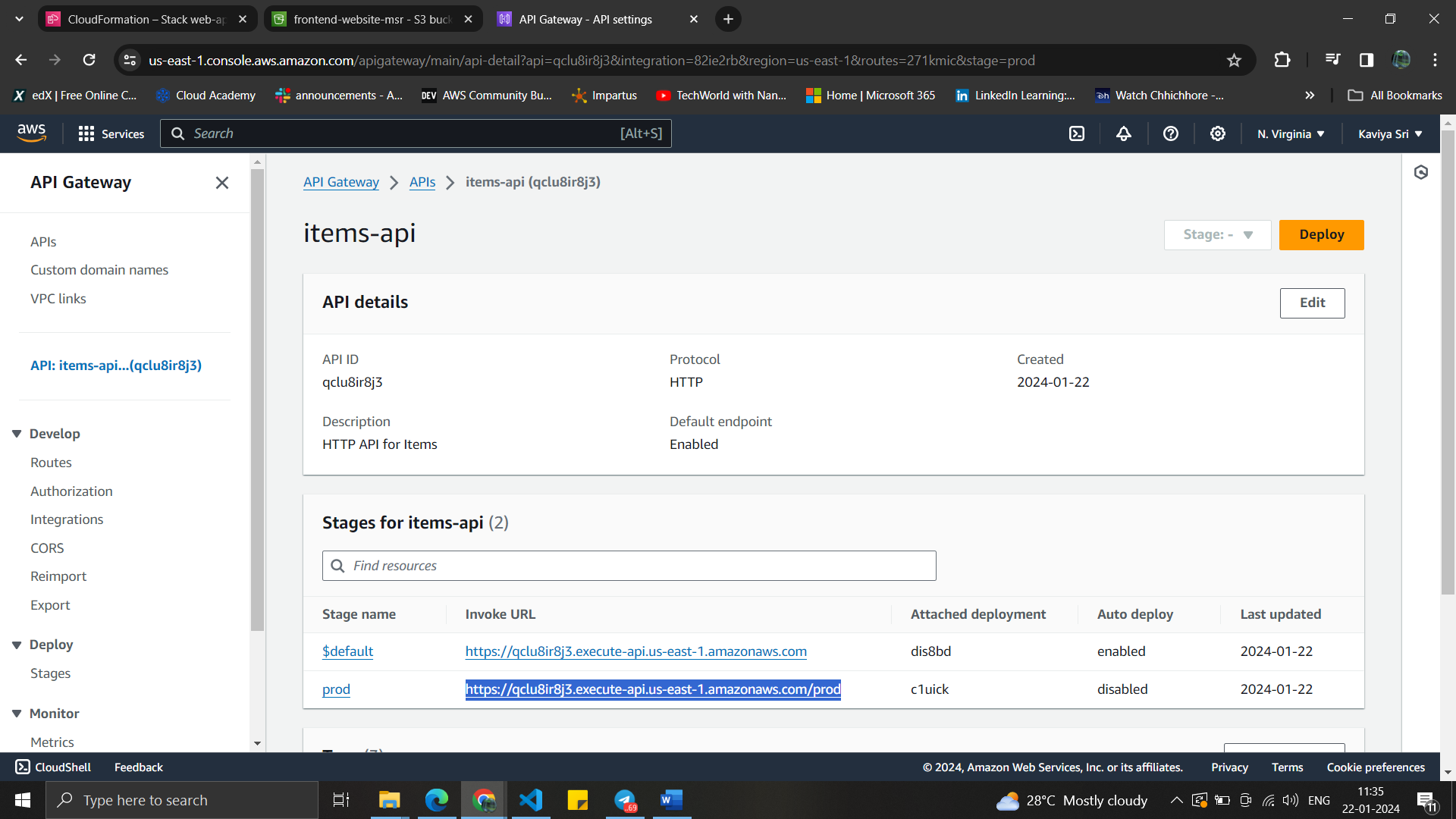


Finally deployed under the “prod” stage



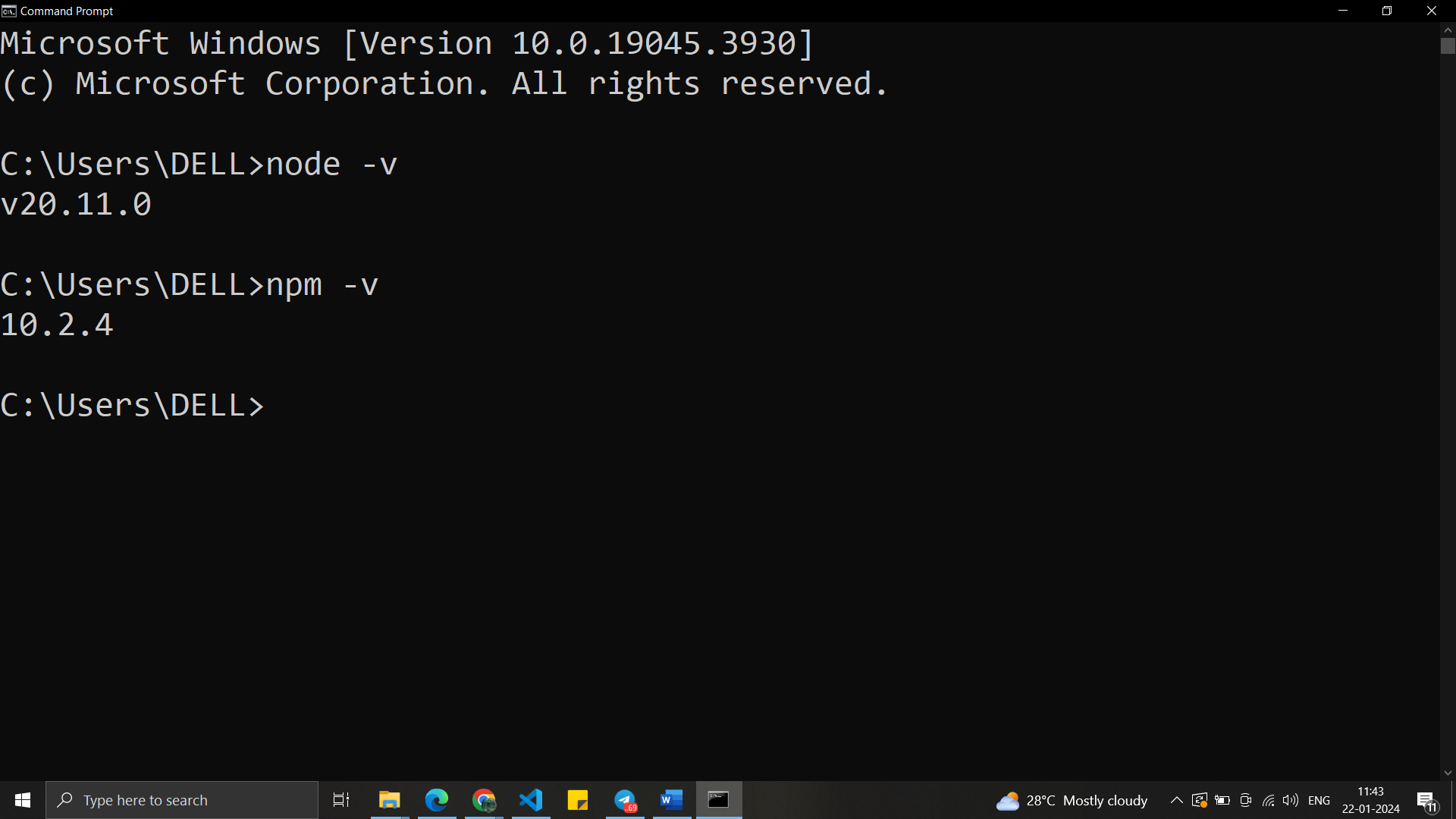


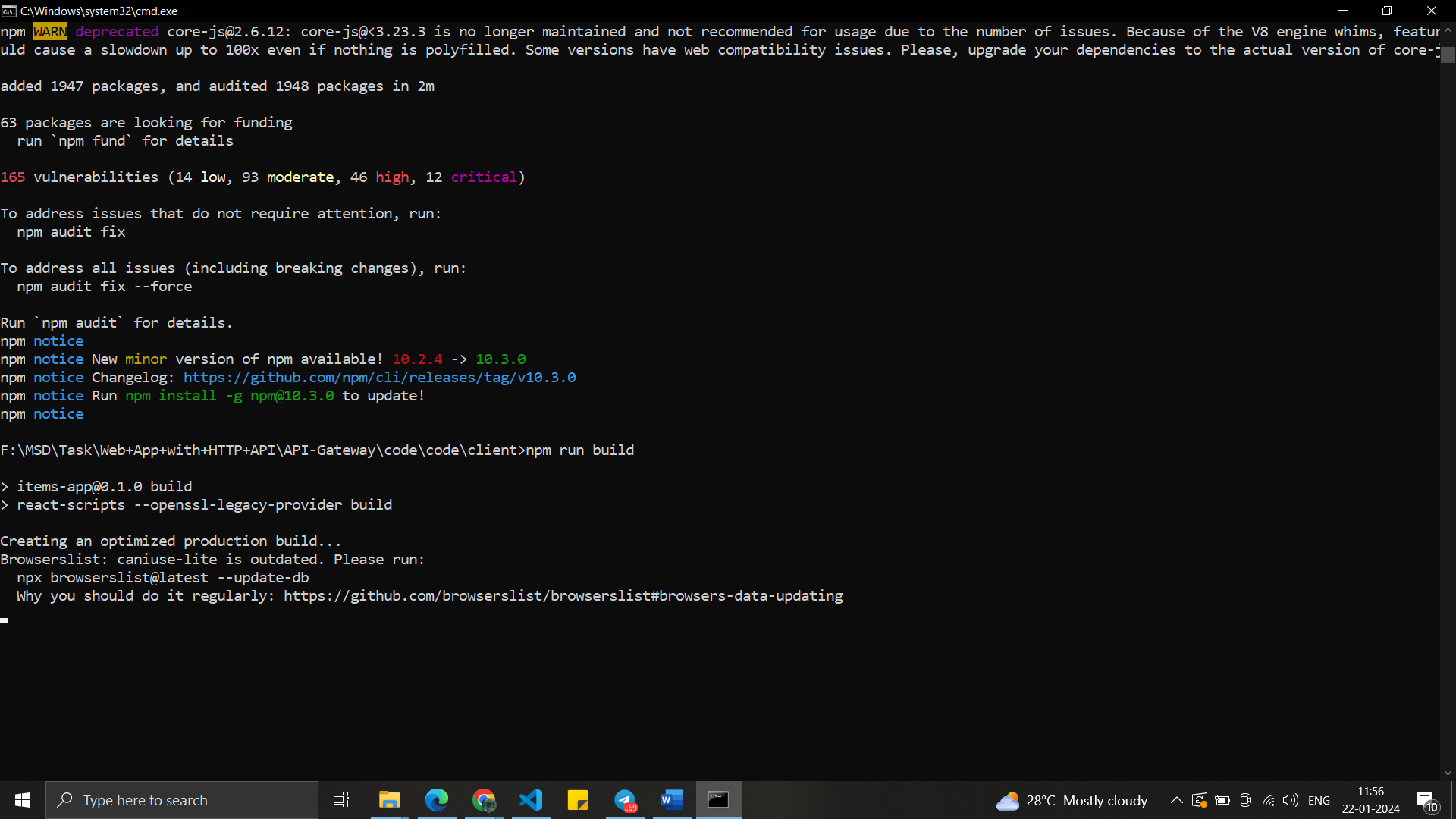
The invoke url is taken from the “items-api” dashboard.



Invoking URL : <https://qclu8ir8j3.execute-api.us-east-1.amazonaws.com/prod>

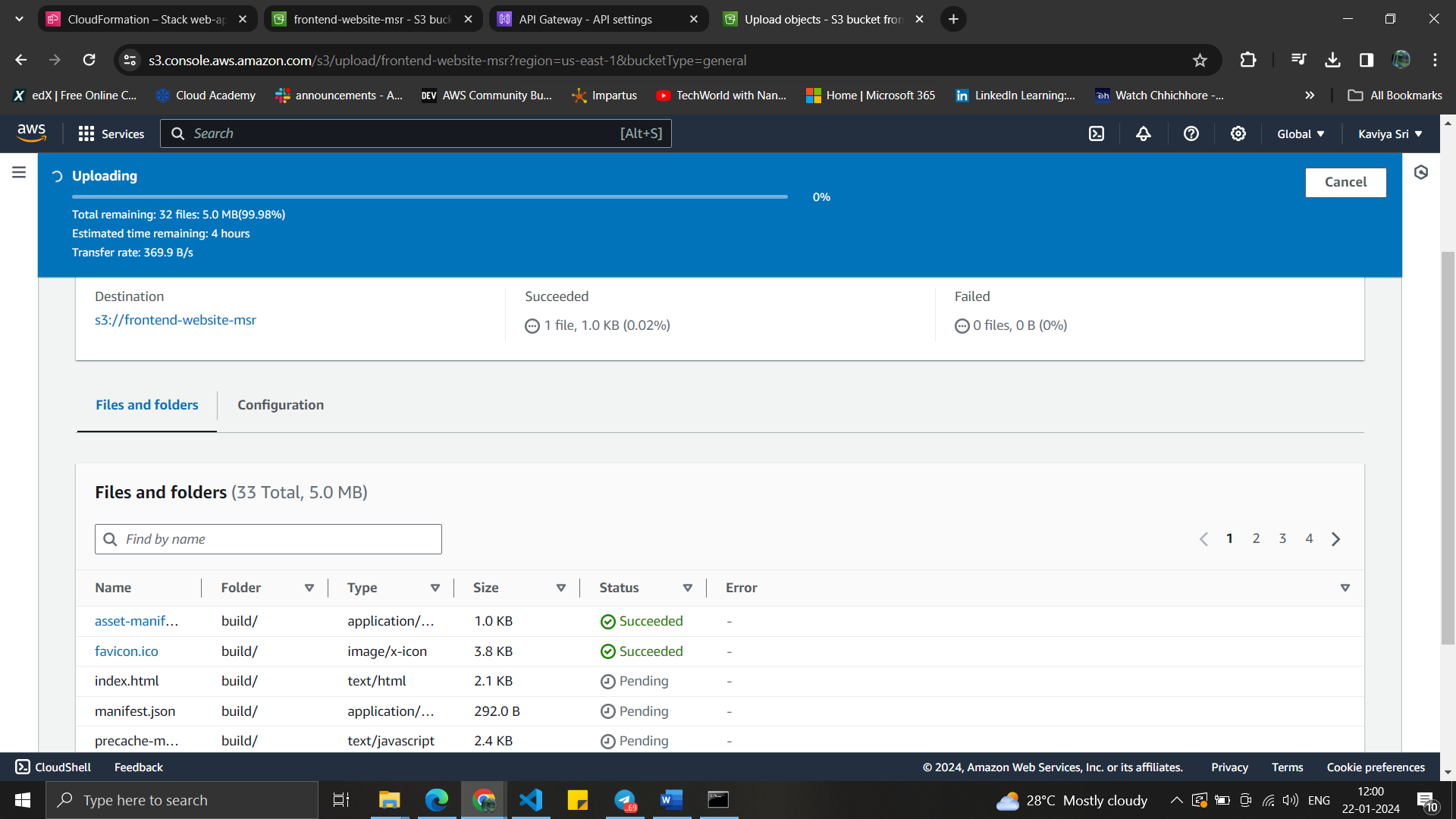
Installed the node js and npm for the frontend website.

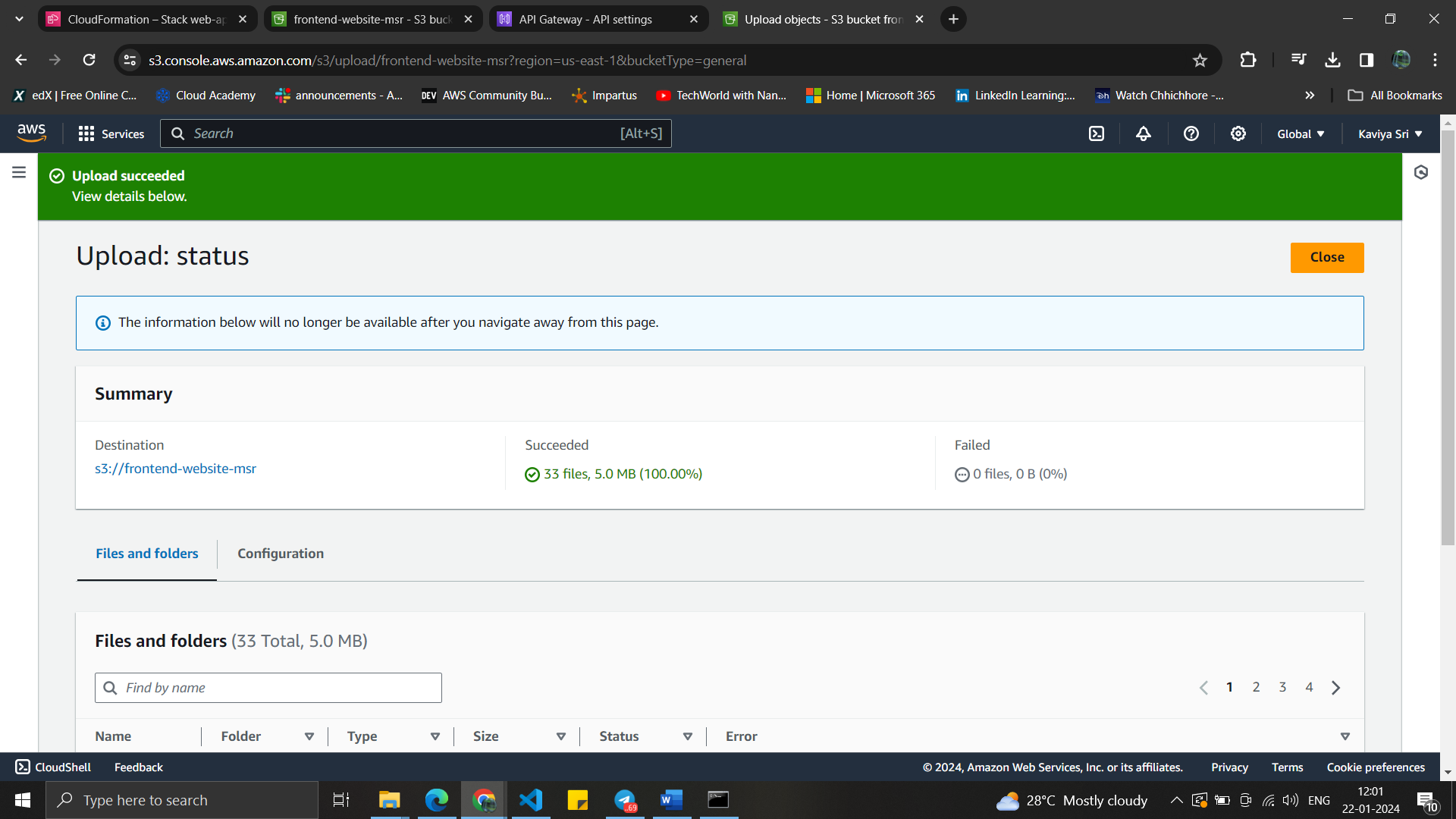




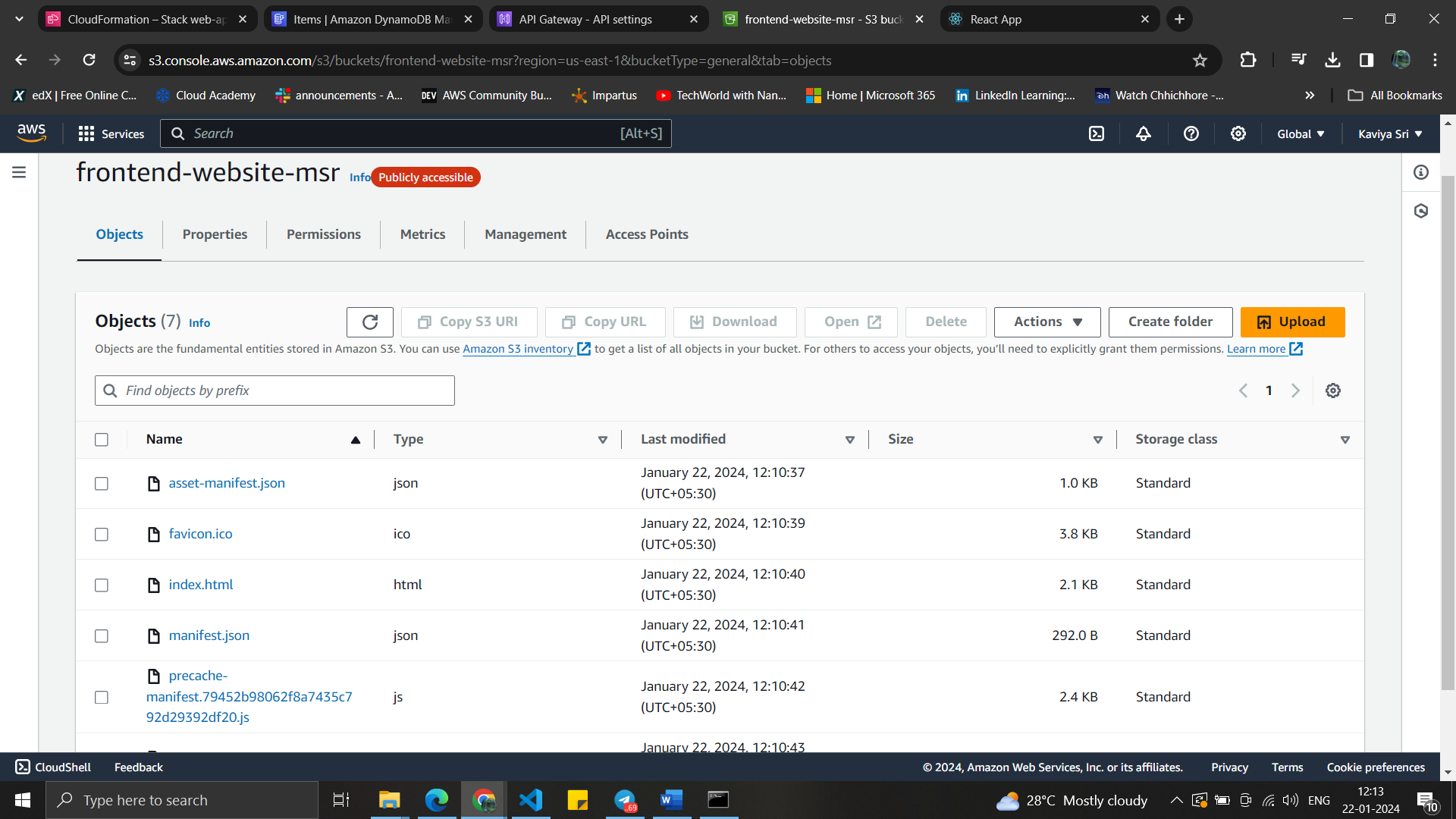


Uploaded all the required objects into the bucket.

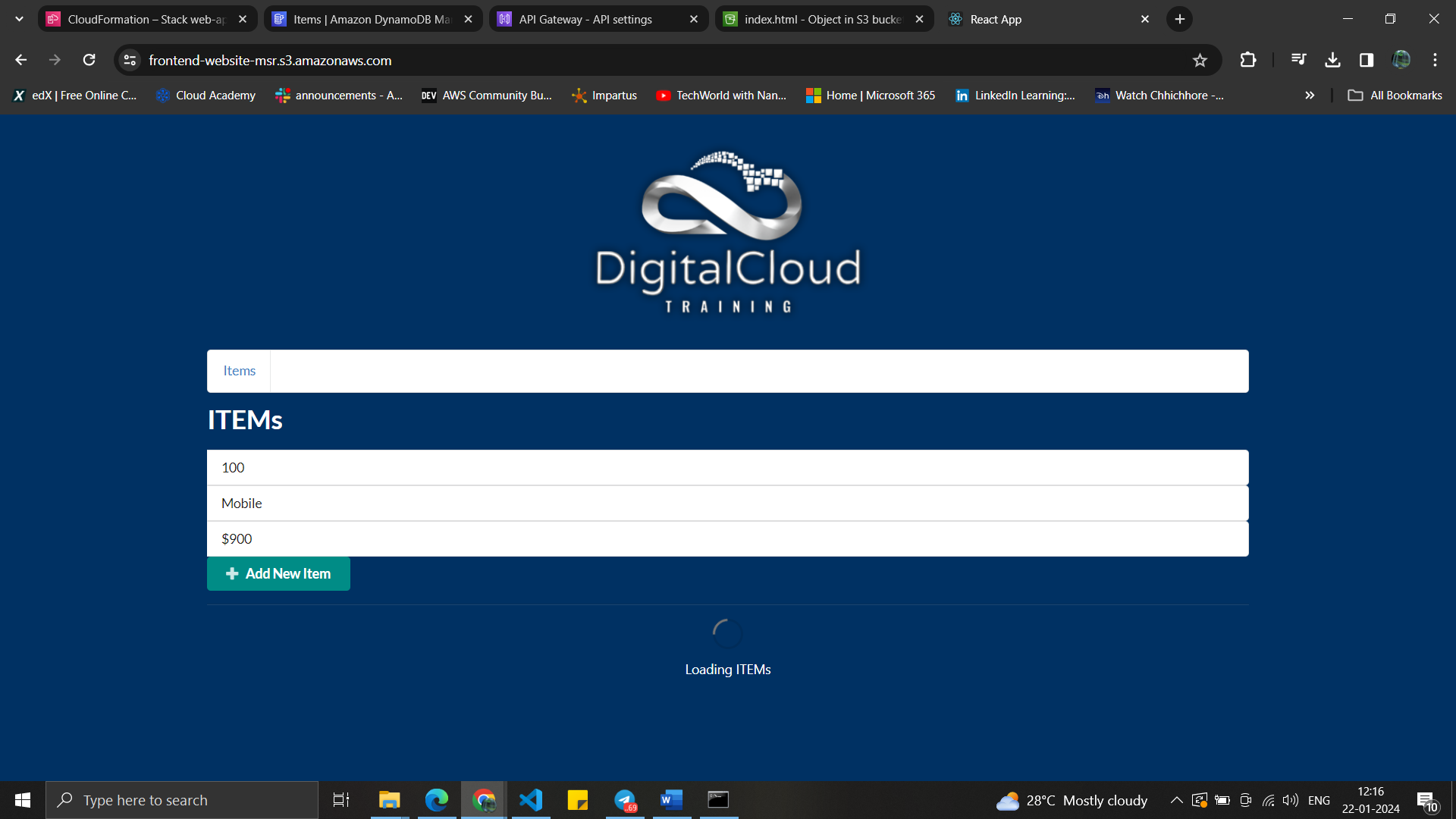




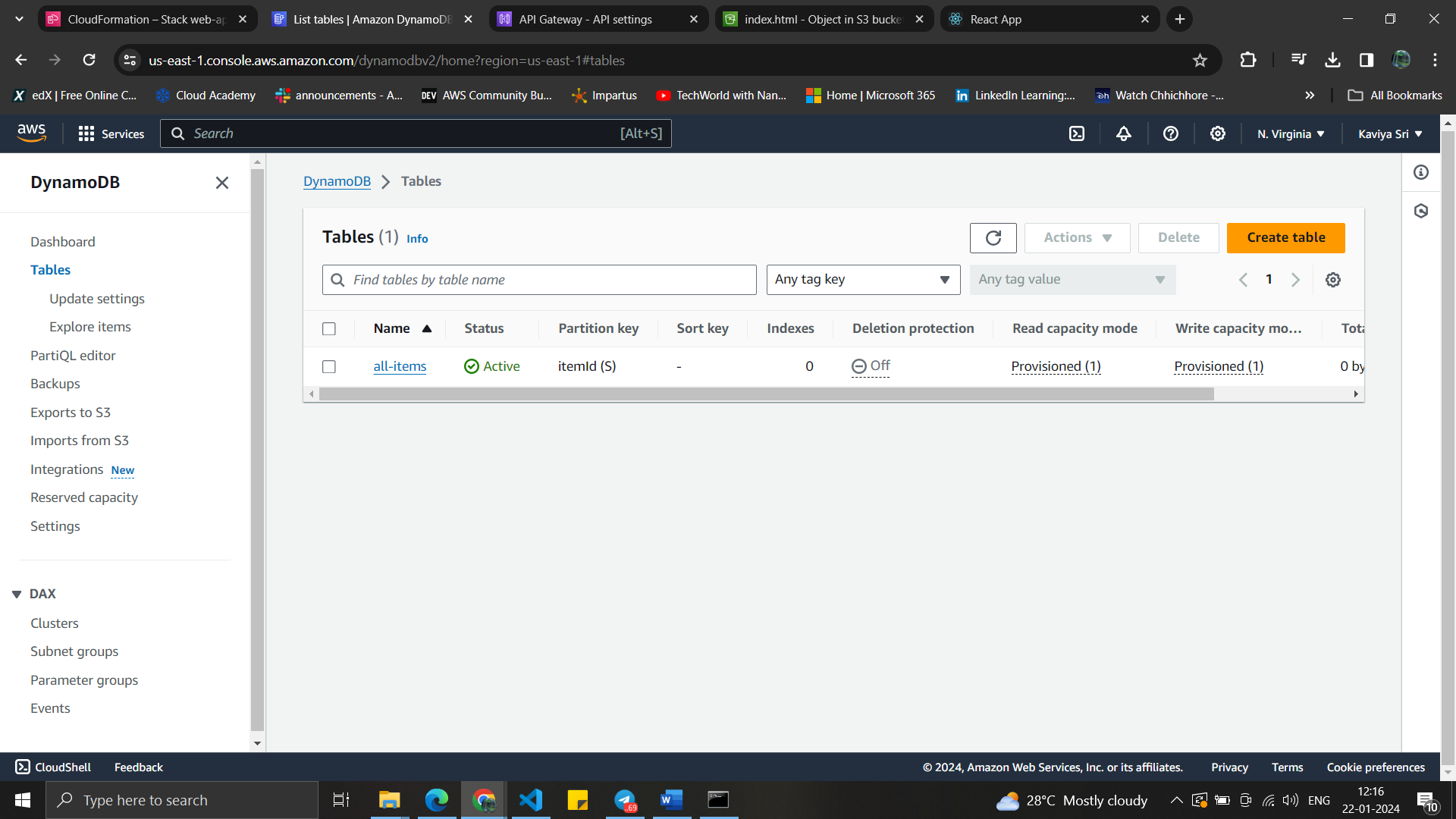
Successfully uploaded all the objects into the bucket for the frontend-website.



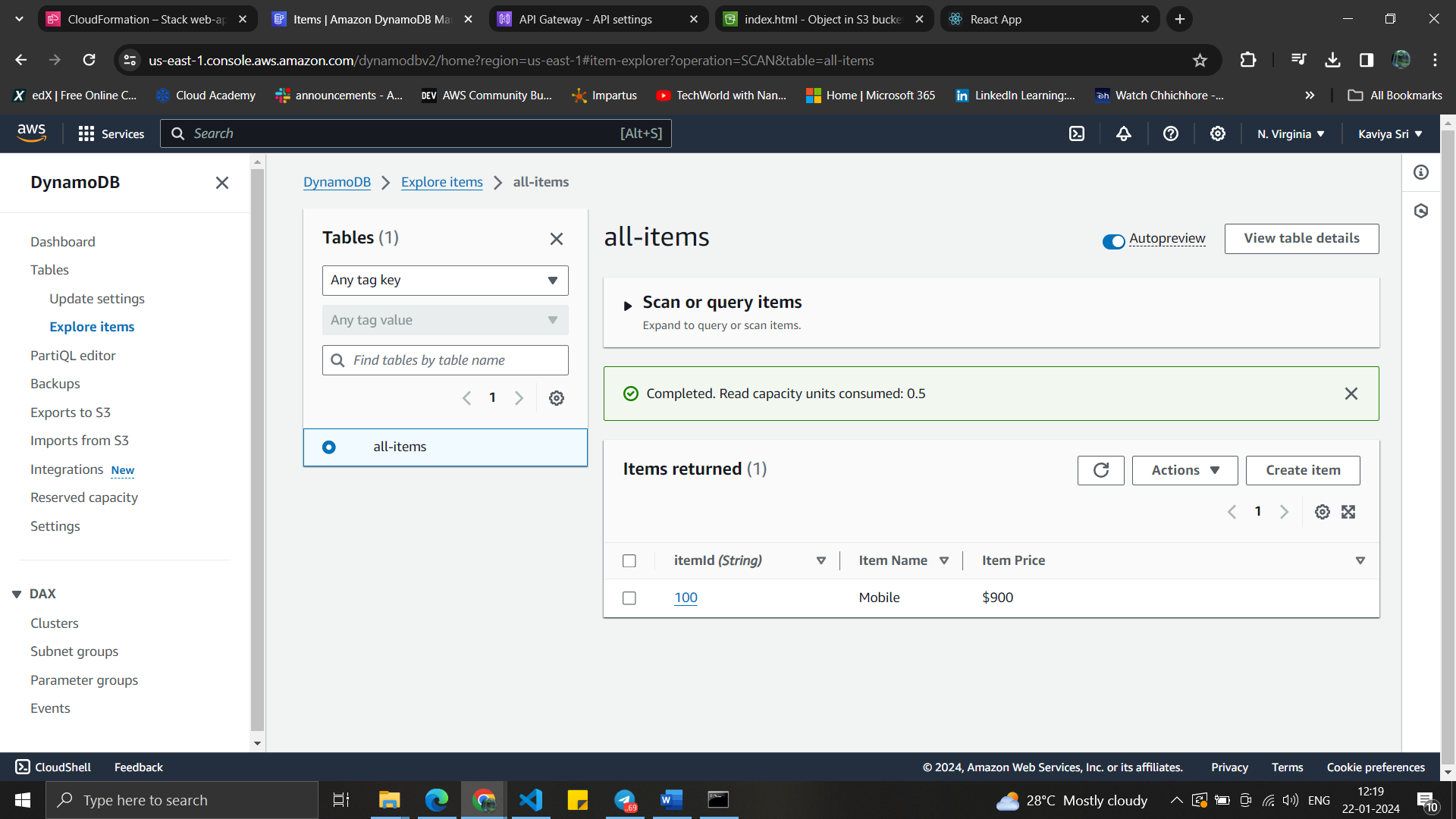
Front end website to upload the items.



The table section has the table named “all-items”.



The items which are added are reflected in the DynamoDB dashboard.



Finally all the created stacks are deleted .

