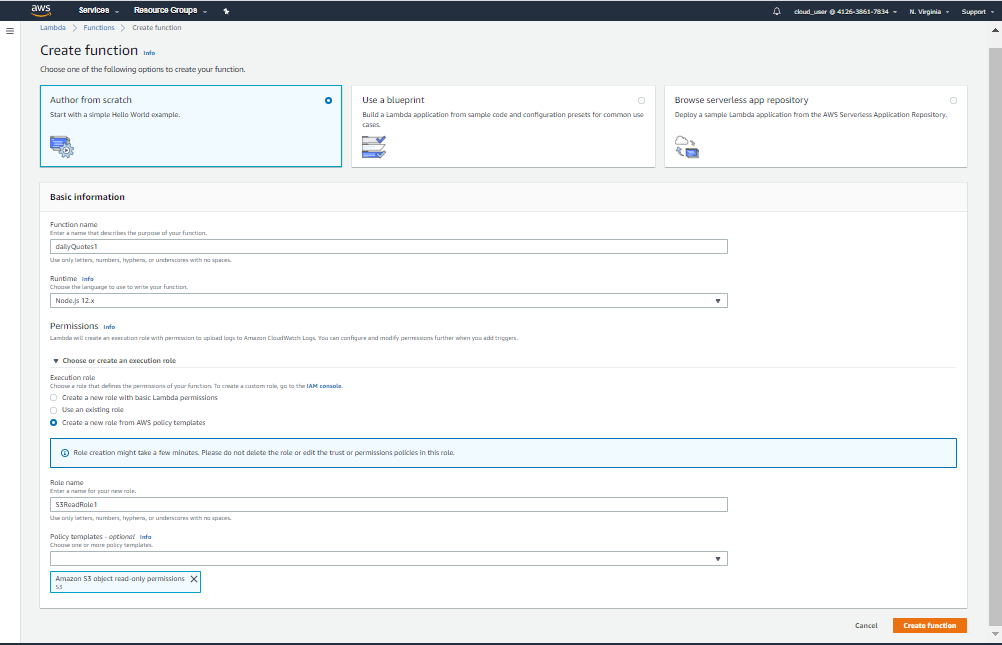
# Testing Objectives

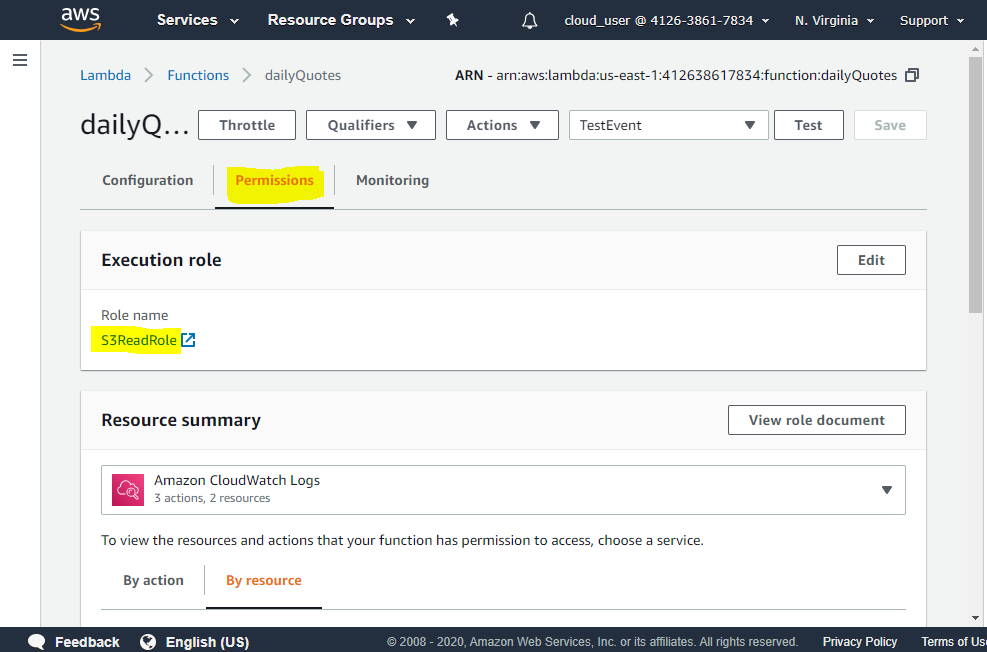
1. Creating a Node + Express JS micro-service using Lambda that randomly serves a quote stored in a file inside S3 bucket
2. Creating an API gateway to provide API endpoint for the Node + Express Service
3. Creating a HTML file , uploaded in S3 bucket to serve as a frontend (which will invoke Lambda function via API Gateway)

# Steps / Screenshots

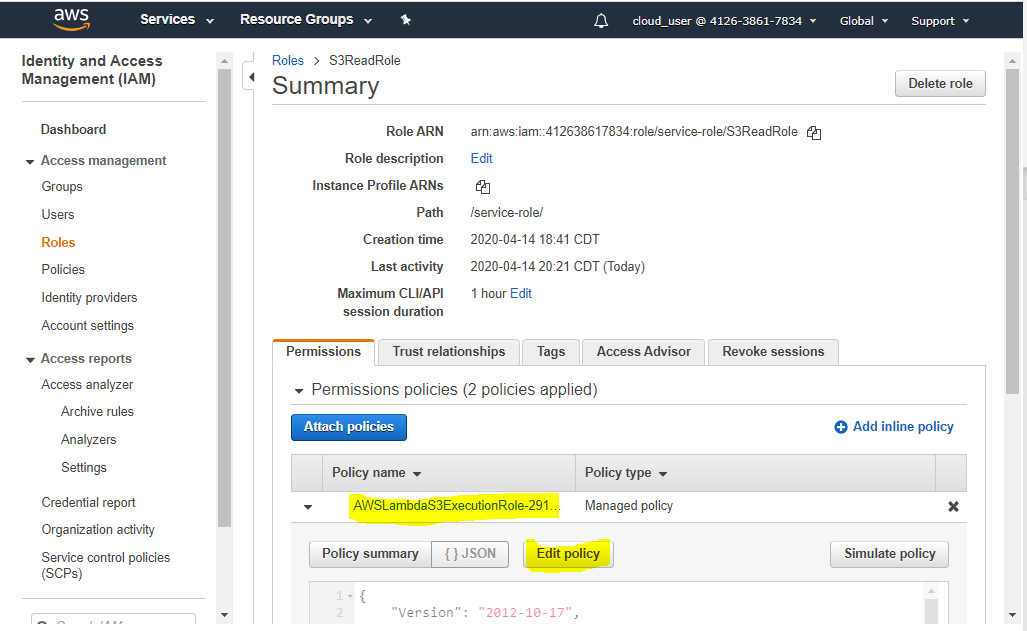
1. Login to AWS Web Console as an Admin user
2. Search for “Lambda” in Services and click on the same
3. From Lambda Dashboard, click on [Create Function] button
4. Select (Author from scratch) option
5. Under [Basic information] section, provide following details
   1. Function name : <some name>
   2. Runtime : Node.js 12.x
6. Under [Choose or create an execution role], select (Create a new role from AWS policy templates) and enter :
   1. Role name : <some name>
   2. Policy templates : Search for “S3” and assign “Amazon S3 object read-only permissions”
7. At this point, the page would look like this :



1. Click on [Create function]
2. Once the function gets created, we need to modify the Policy a little bit so that the function can have all S3 access
3. Go to [Permissions] tab & click on the Role Name which will take you to IAM page



1. In IAM page, expand “AWSLambdaS3ExecutionRole….” & then click on [Edit Policy] button



1. Copy/paste following JSON in JSON tab of edit policy window

{

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

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"s3:GetLifecycleConfiguration",

"s3:GetBucketTagging",

"s3:GetInventoryConfiguration",

"s3:GetObjectVersionTagging",

"s3:GetBucketLogging",

"s3:ListBucket",

"s3:GetAccelerateConfiguration",

"s3:GetBucketPolicy",

"s3:GetObjectVersionTorrent",

"s3:GetObjectAcl",

"s3:GetEncryptionConfiguration",

"s3:GetBucketObjectLockConfiguration",

"s3:GetBucketRequestPayment",

"s3:GetAccessPointPolicyStatus",

"s3:GetObjectVersionAcl",

"s3:GetObjectTagging",

"s3:GetMetricsConfiguration",

"s3:GetBucketPublicAccessBlock",

"s3:GetBucketPolicyStatus",

"s3:GetObjectRetention",

"s3:GetBucketWebsite",

"s3:GetBucketVersioning",

"s3:GetBucketAcl",

"s3:GetObjectLegalHold",

"s3:GetBucketNotification",

"s3:GetReplicationConfiguration",

"s3:GetObject",

"s3:GetObjectTorrent",

"s3:GetBucketCORS",

"s3:GetAnalyticsConfiguration",

"s3:GetObjectVersionForReplication",

"s3:GetBucketLocation",

"s3:GetAccessPointPolicy",

"s3:GetObjectVersion"

],

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

},

{

"Sid": "VisualEditor1",

"Effect": "Allow",

"Action": [

"s3:GetAccessPoint",

"s3:GetAccountPublicAccessBlock",

"s3:ListAllMyBuckets"

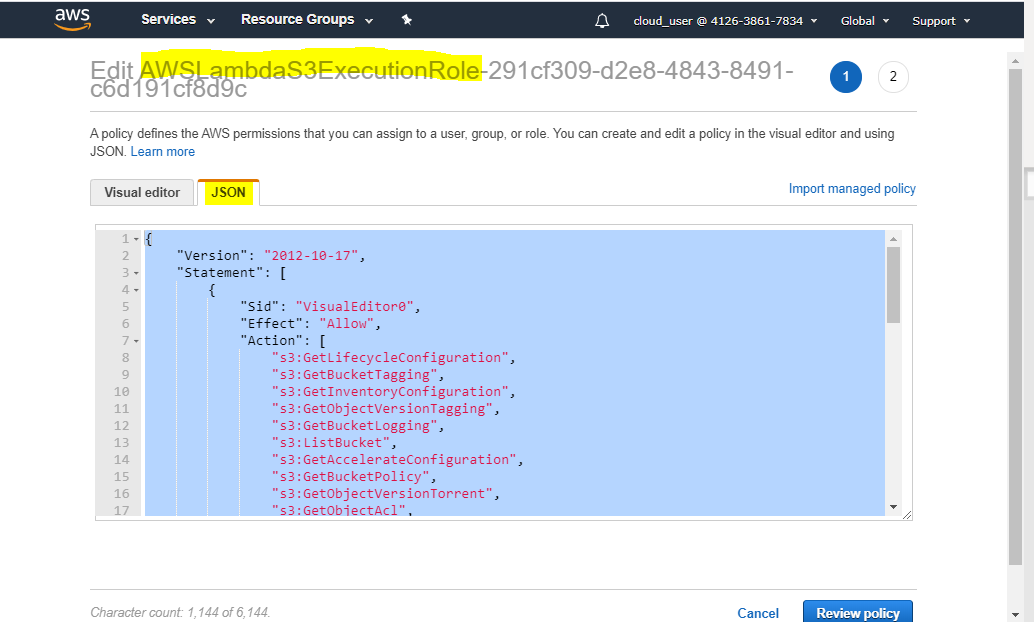
],

"Resource": "\*"

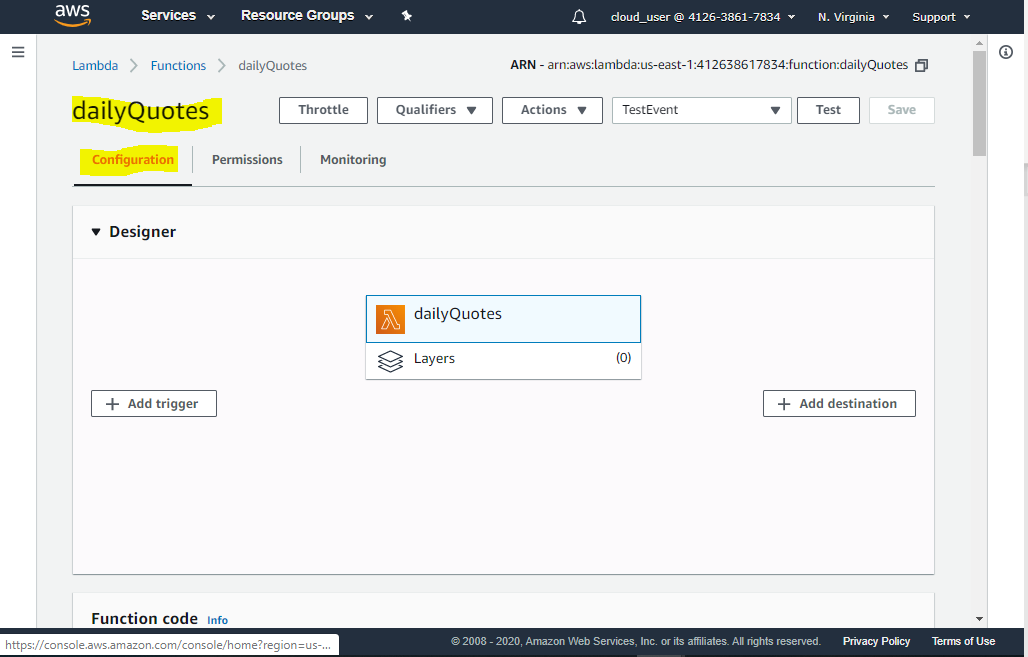
}

]

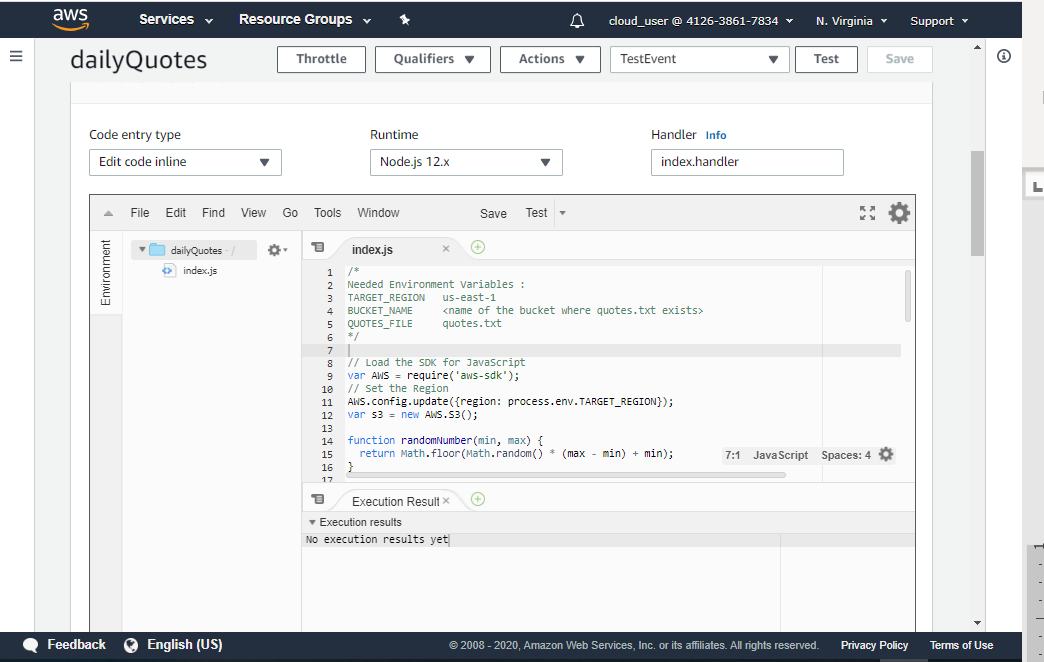
}



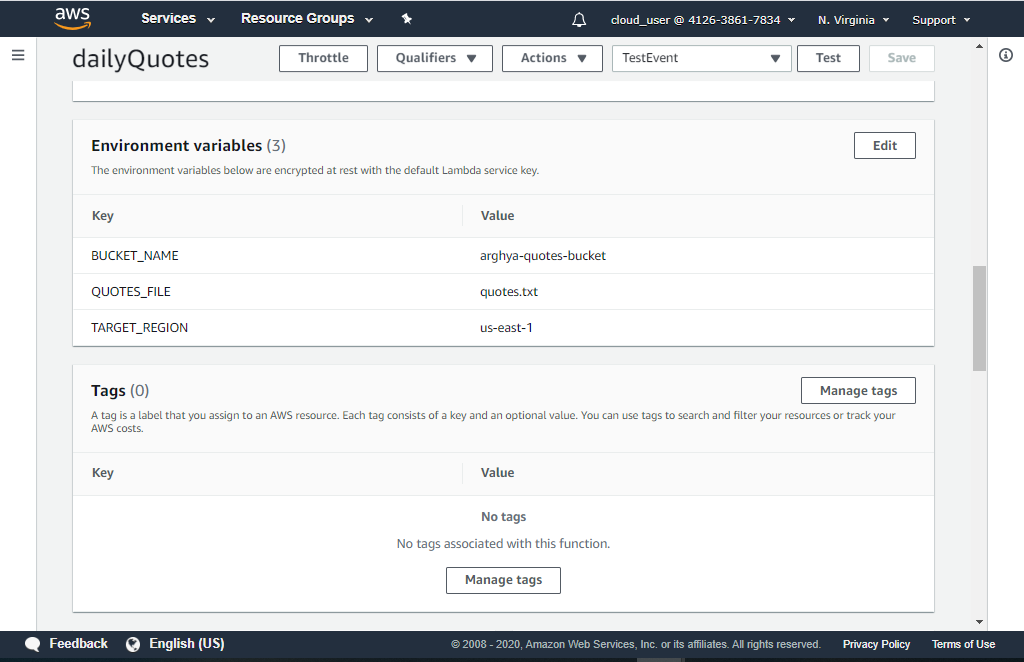
1. Click on [Review Policy] & then [Save changes] buttons
2. This ensures Lambda function has all the read privileges in S3
3. Now, go back to Lambda function page & navigate to [Configuration] tab



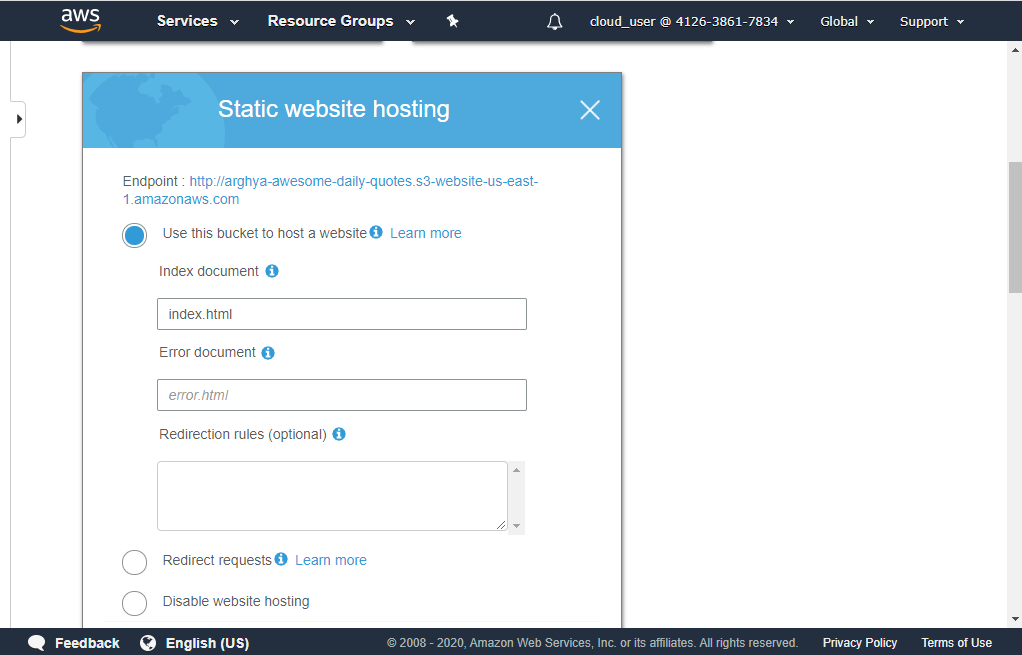
1. Inside inline code editor, copy/paste the code in “index.js” file. This lambda function simply lists files inside a given bucket 🡪 checks whether there is a file named “quotes.txt” 🡪 if so, it reads the content of the file 🡪 splits the content by lines 🡪 returns a random line or quote



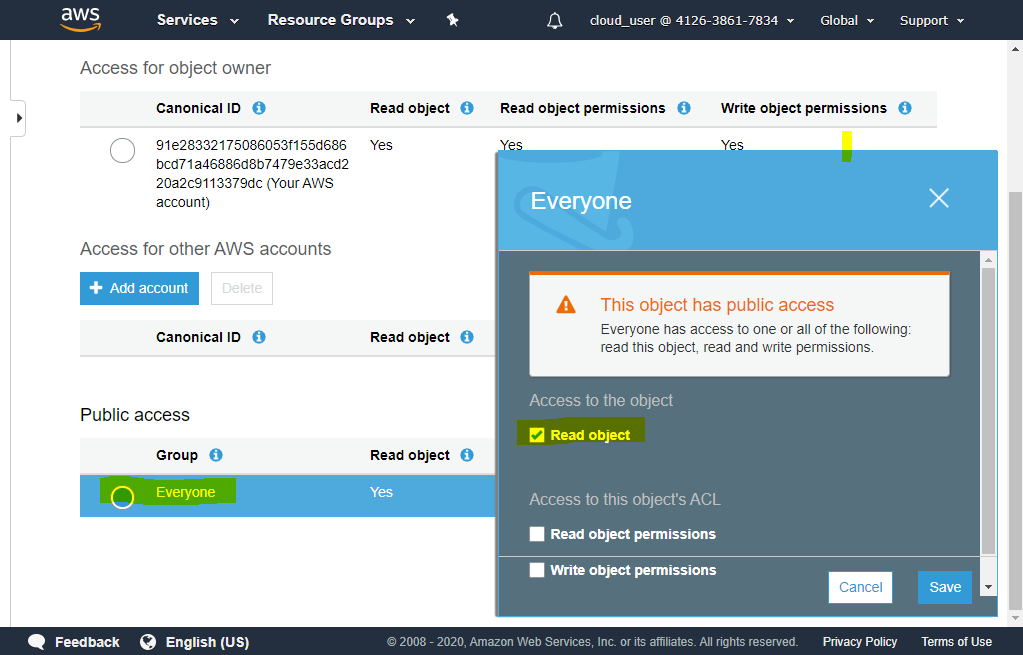
1. Under [Environment variables section], add following key, value pairs as appropriate for you
   1. BUCKET\_NAME : <YOUR\_NAME>-quotes-bucket (will be created in next step)
   2. QUOTES\_FILE : quotes.txt (will be uploaded in next step)
   3. TARGET\_REGION : us-east-1



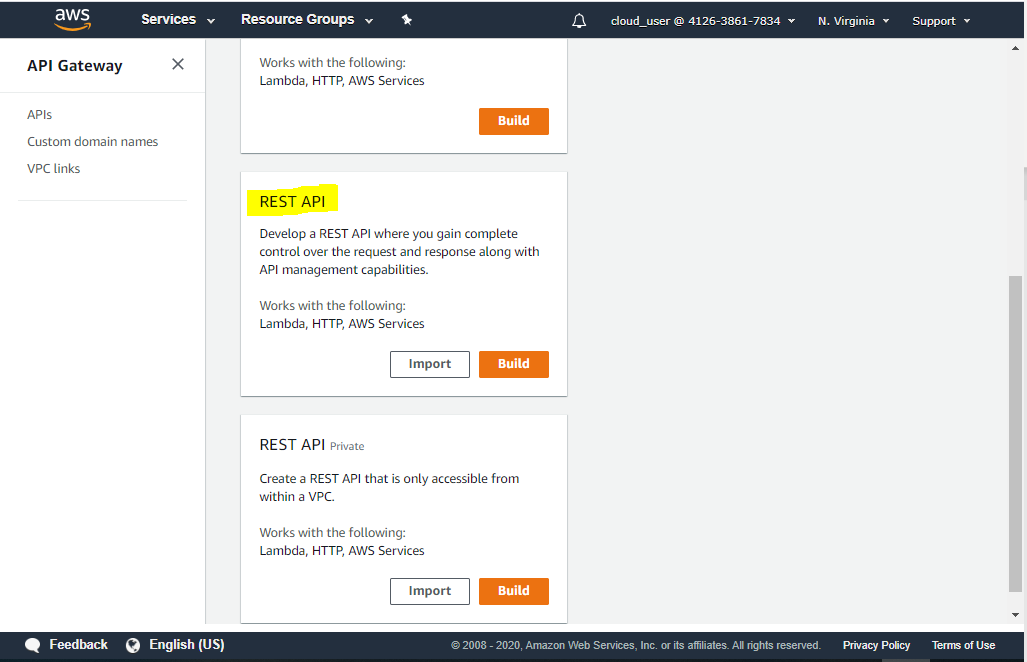
1. Now, go to AWS web console & search for S3 service page & create two buckets:
   1. <YOUR\_NAME>-quotes-bucket : Keep the default permissions, do not change anything. This bucket will contain the quotes.txt file
   2. <YOUR\_NAME>-awesome-daily-quotes : For this, **uncheck “Block public access” checkbox**. This bucket will host the quotes web page, and hence it must be public. Once this bucket is created, go to [Properties] tab 🡪 Click on [Static website hosting] 🡪 Click on “Use this bucket to host a website” 🡪 enter “Index document” as “index.html”



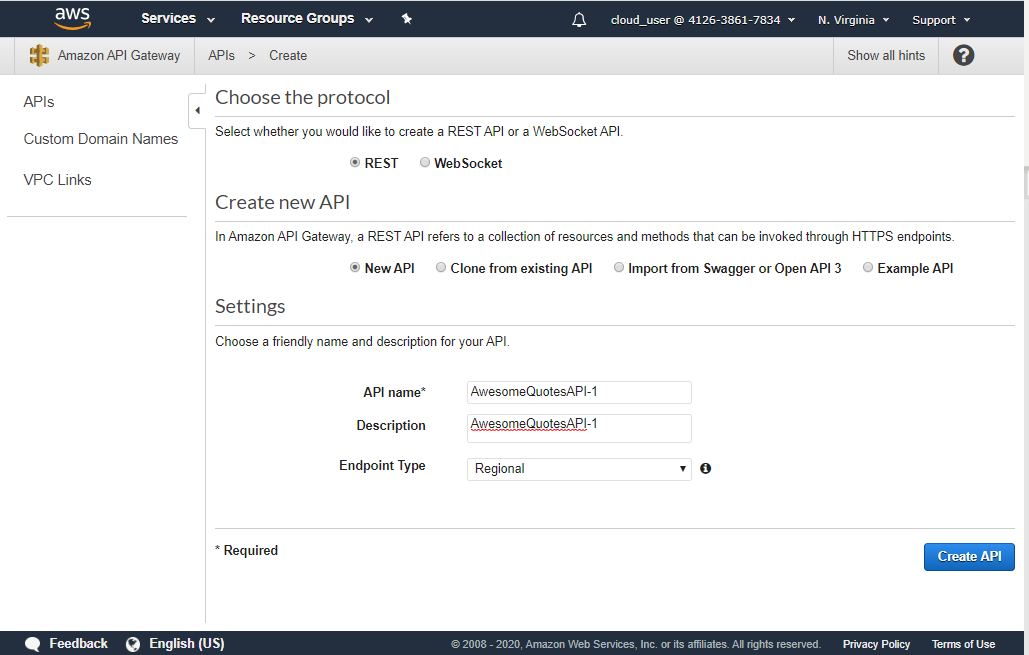
1. Now, upload quotes.txt file in <YOUR\_NAME>-quotes-bucket. No need to change any configuration
2. Now, upload index.html in <YOUR\_NAME>-awesome-daily-quotes bucket. No need to change any configuration at this stage. But in next stage we will give public read access
3. Once the file is uploaded, click on it 🡪 go to [Permissions” tab 🡪 go to “Public access” section 🡪 Click on the radio button beside “Everyone” 🡪 and check on [Read object] checkbox



1. We would need to change this file content a bit, but will come to that at the end
2. Go to AWS web console and search for “API Gateway”
3. Click on [Create API] button
4. In next page, select to build REST API

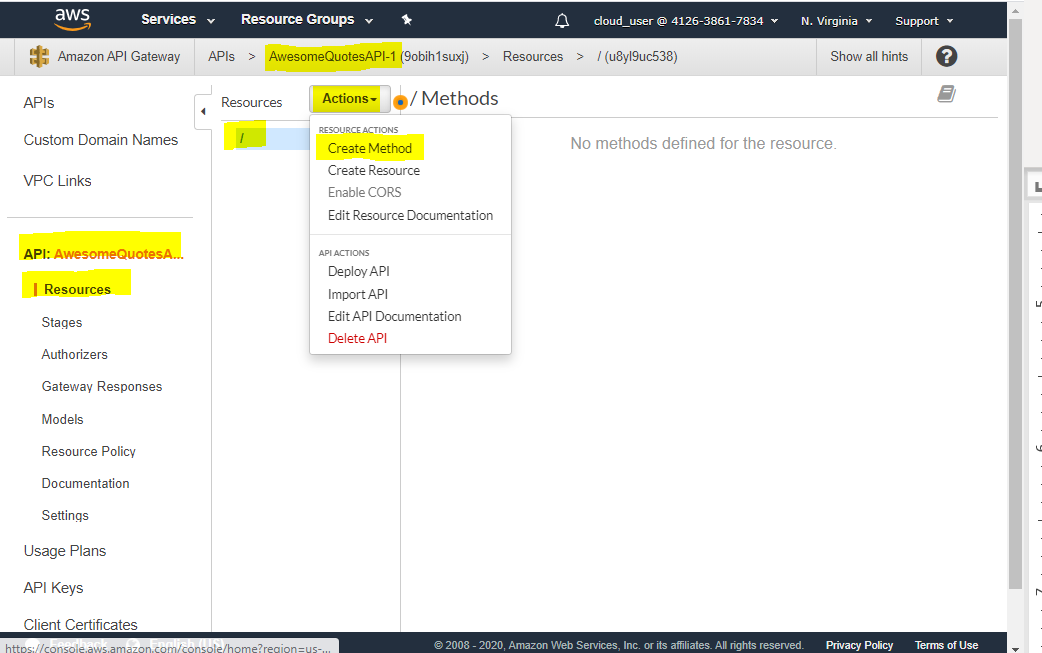


1. Enter following details :
   1. API name : <some name>
   2. Description: <some description>
   3. Endpoint type : Regional

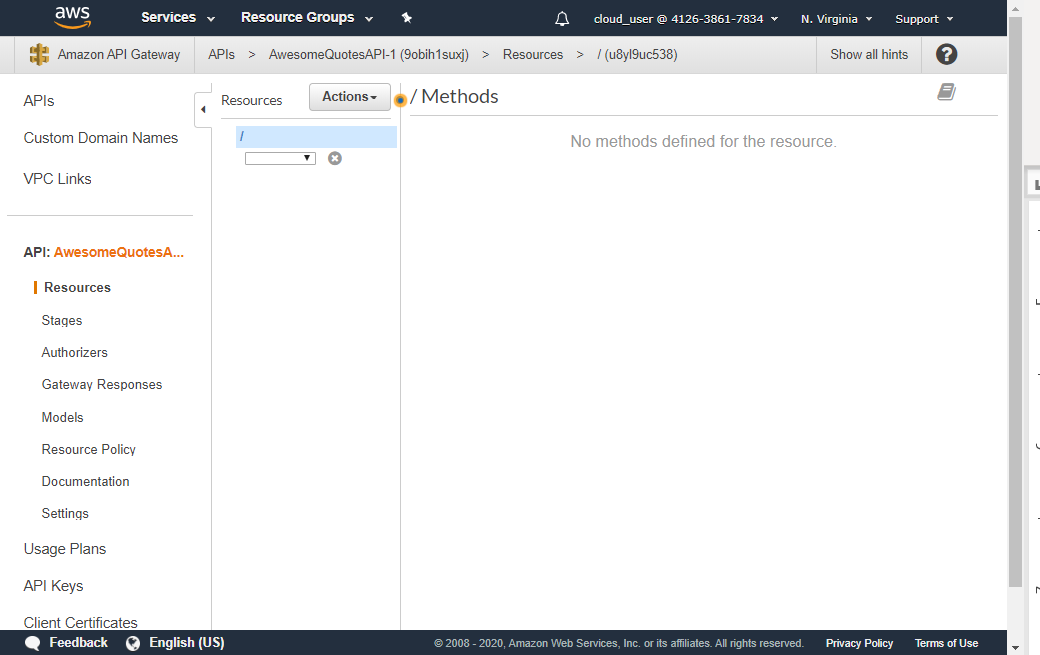


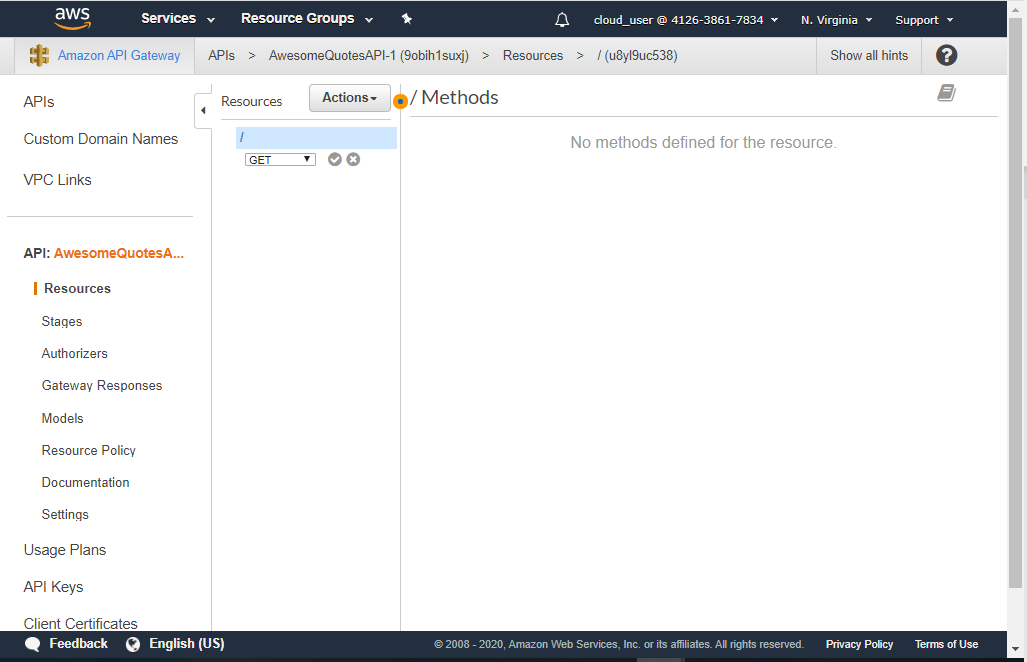
Click on [Create API] button

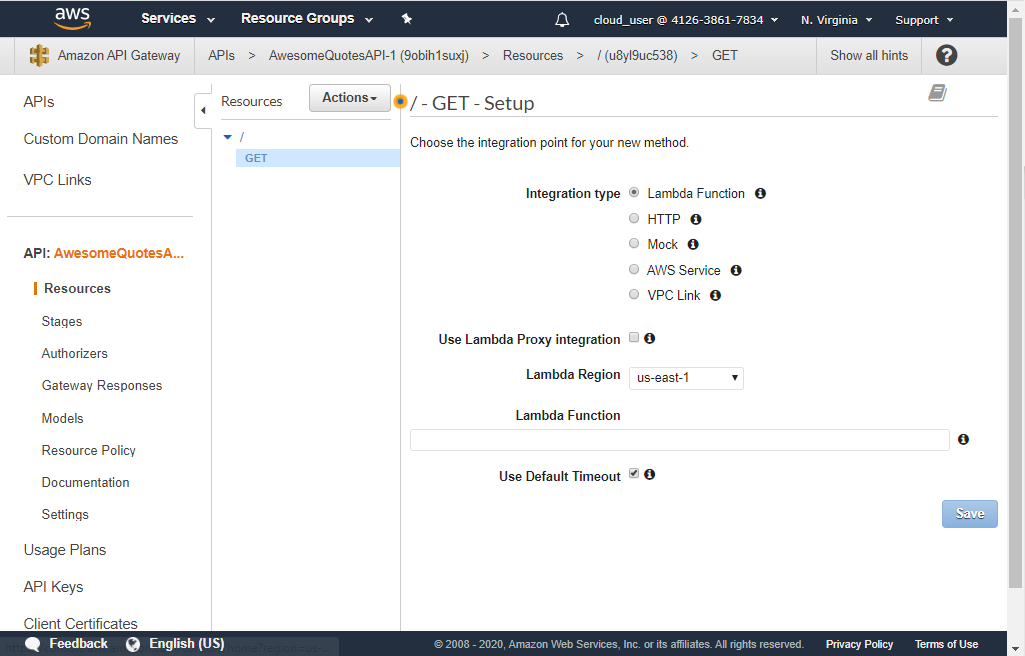
1. Ensure that “/” path is selected & then select [Actions] 🡪 [Create Method]



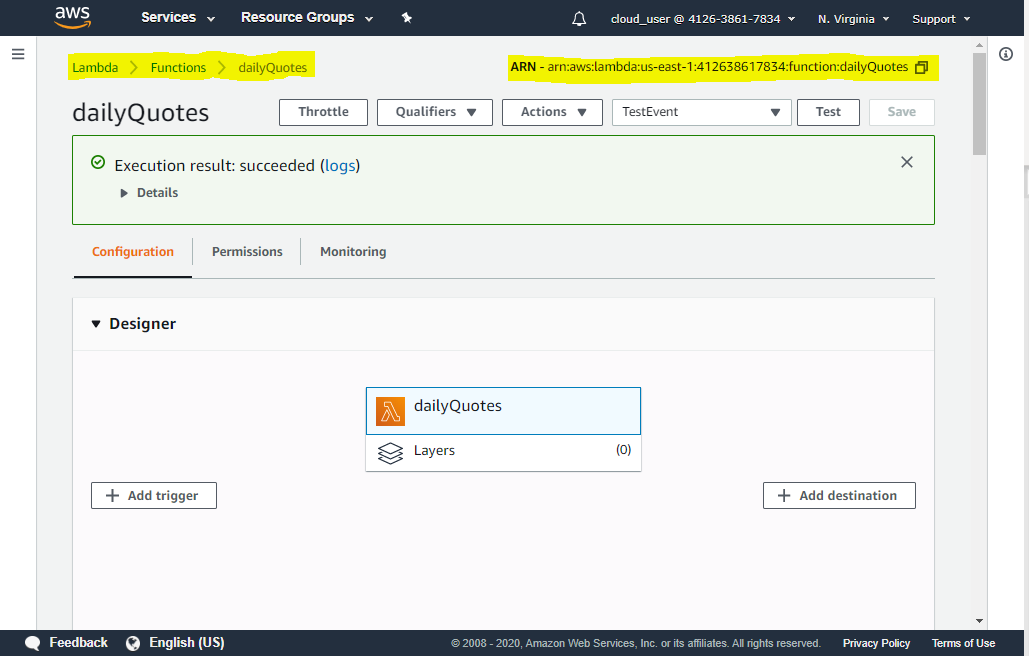
1. From the drop down, select “GET”, then click on the check icon beside the method

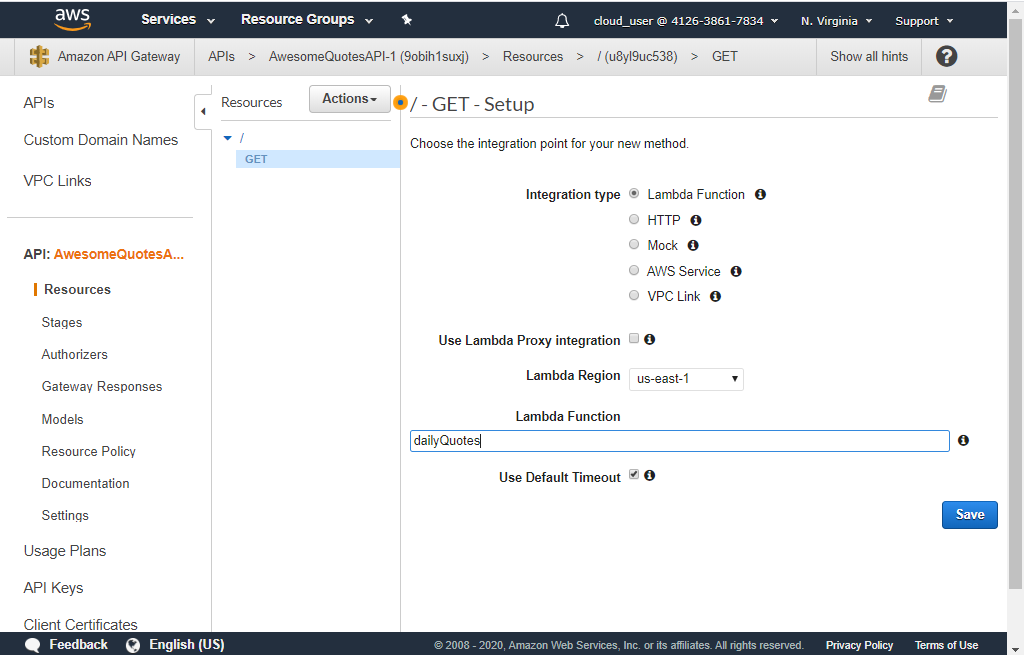






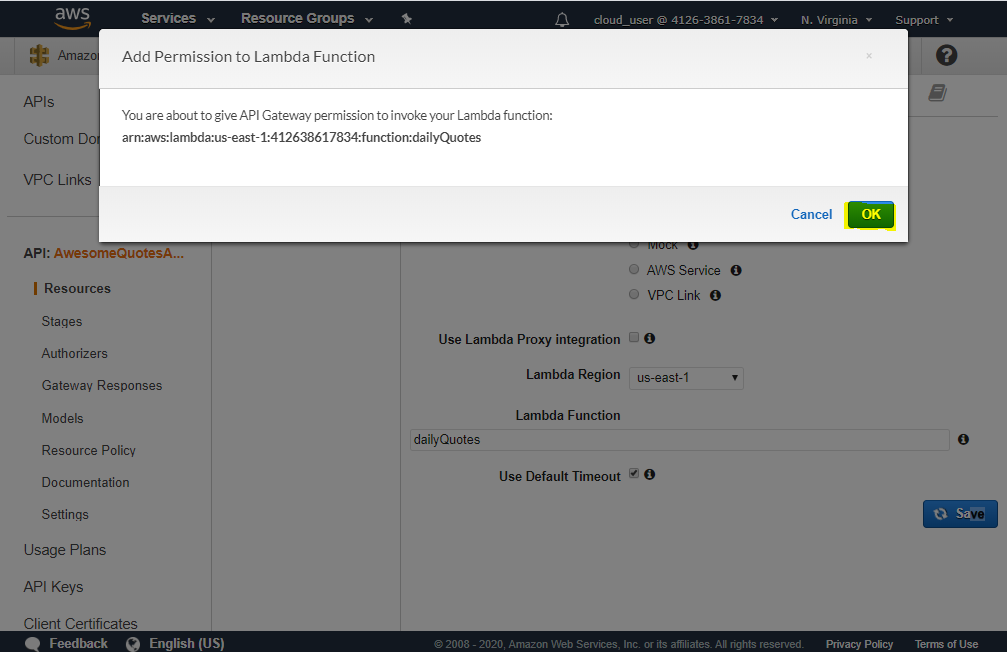
1. In the current screen, you need to enter the ARN (Amazon Resource Name) of the Lambda function created in previous step



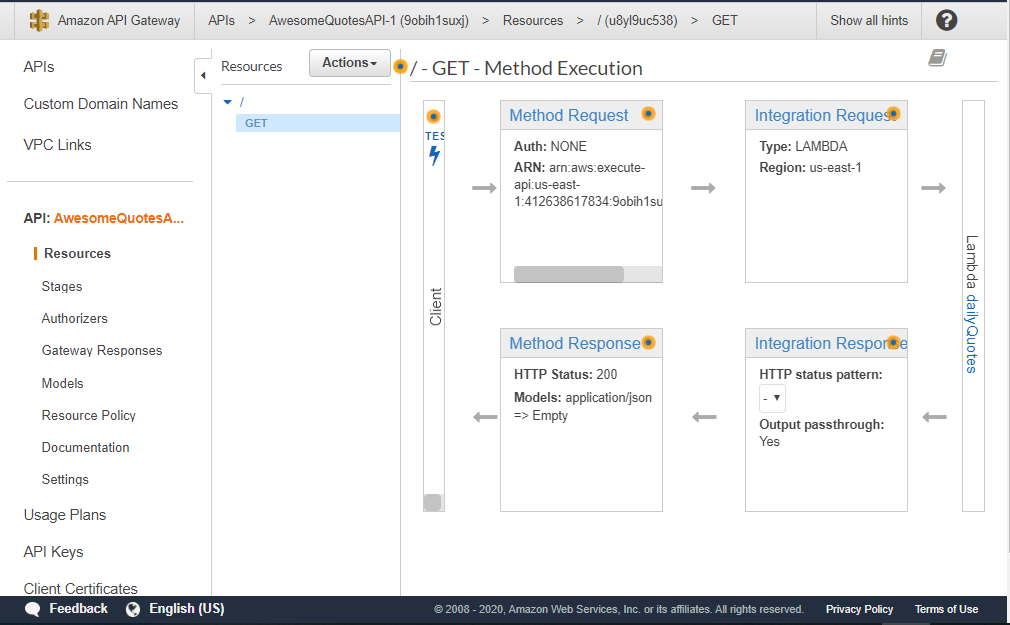


Note : in “Lambda Function” text box, the whole ARN needs to be placed & it would automatically replace the ARN with the name. So, don’t get confused by the screenshot above

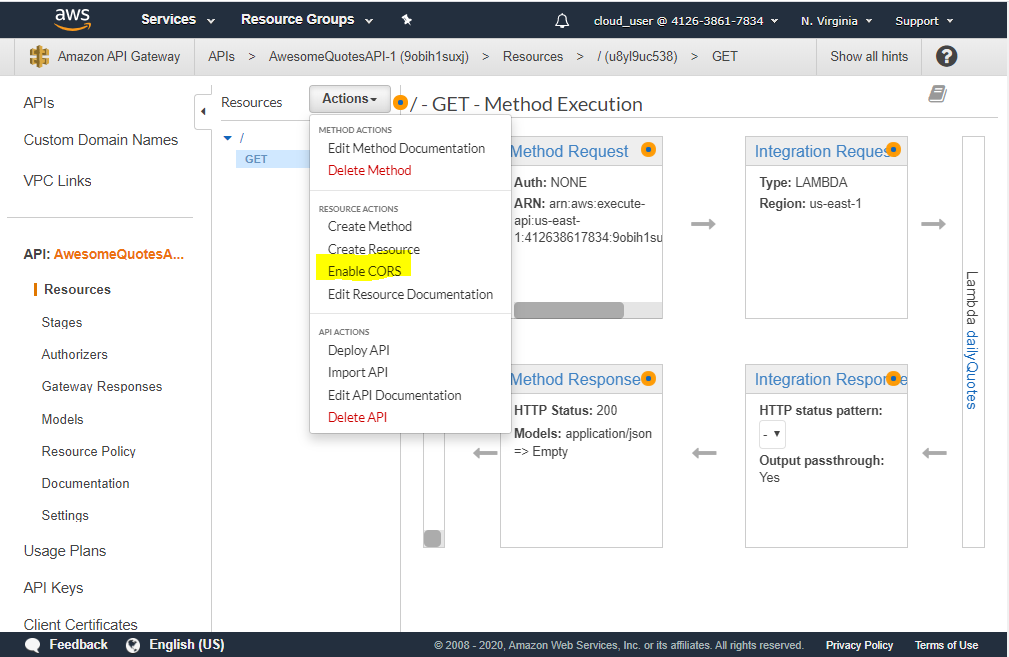
1. Click on [Save]
2. When the popup appears, click on [Ok]



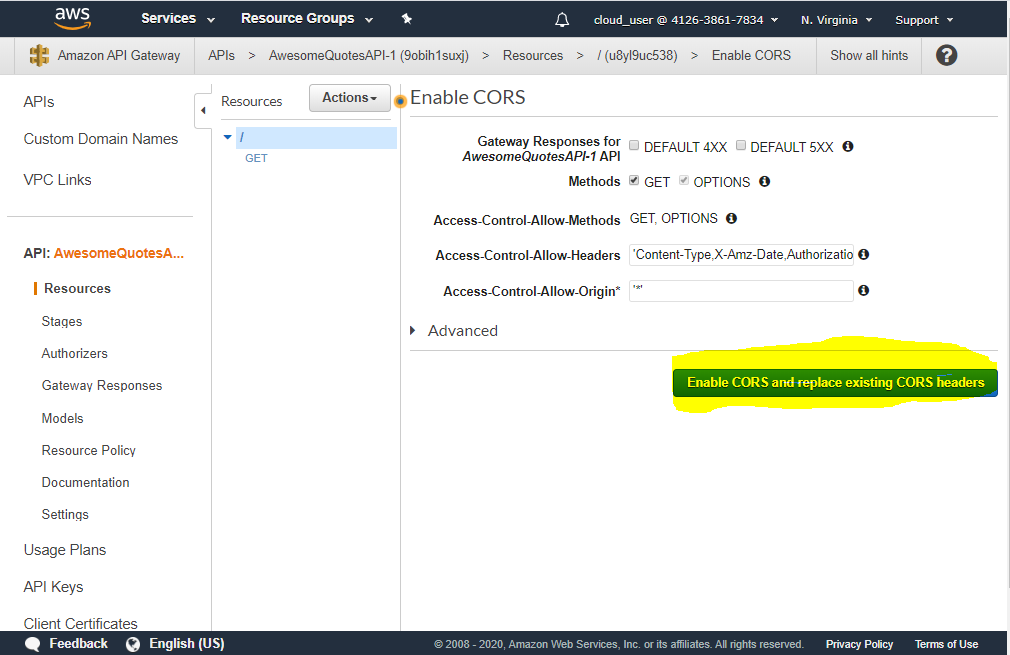
1. Following page appears



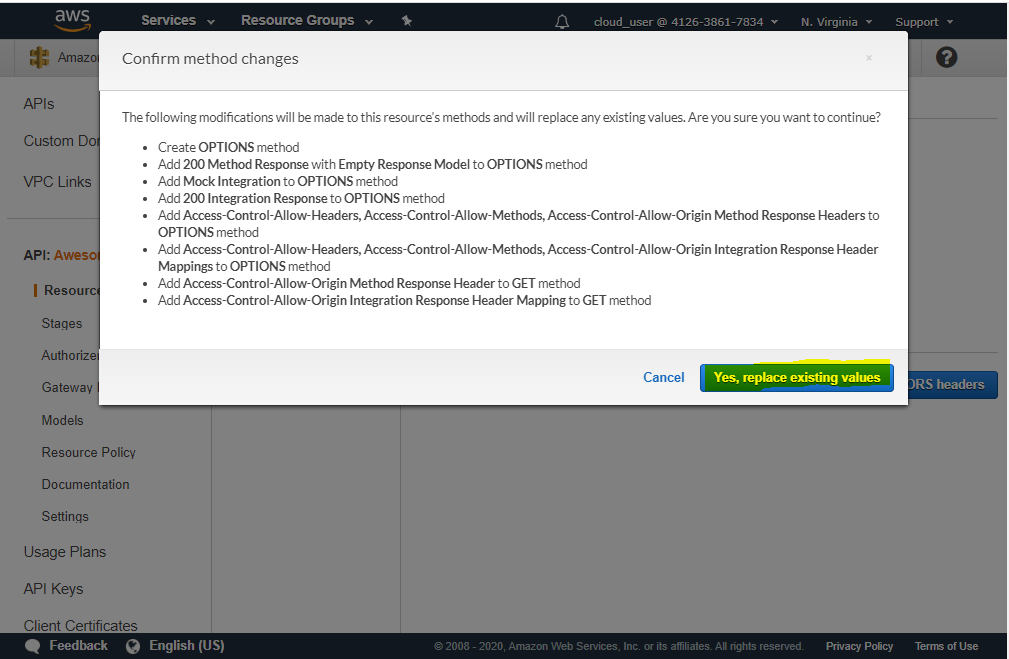
1. Ensuring that the GET endpoint is still selected, chose Actions 🡪 Enable CORS

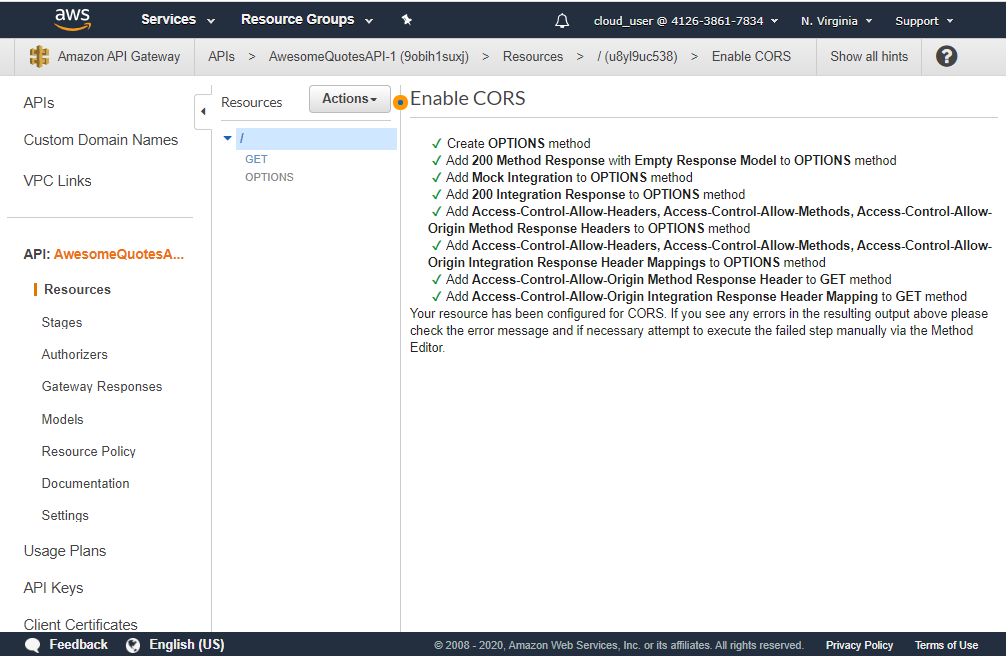


1. Then click on [Enable CORS ….] button

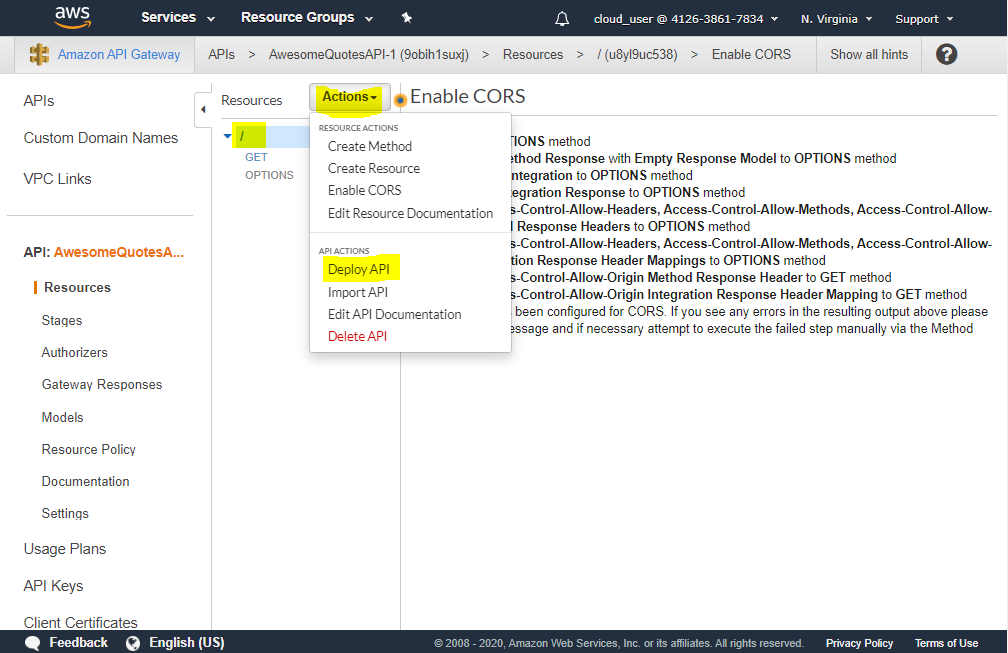


1. In the popup, click on [Yes,…]

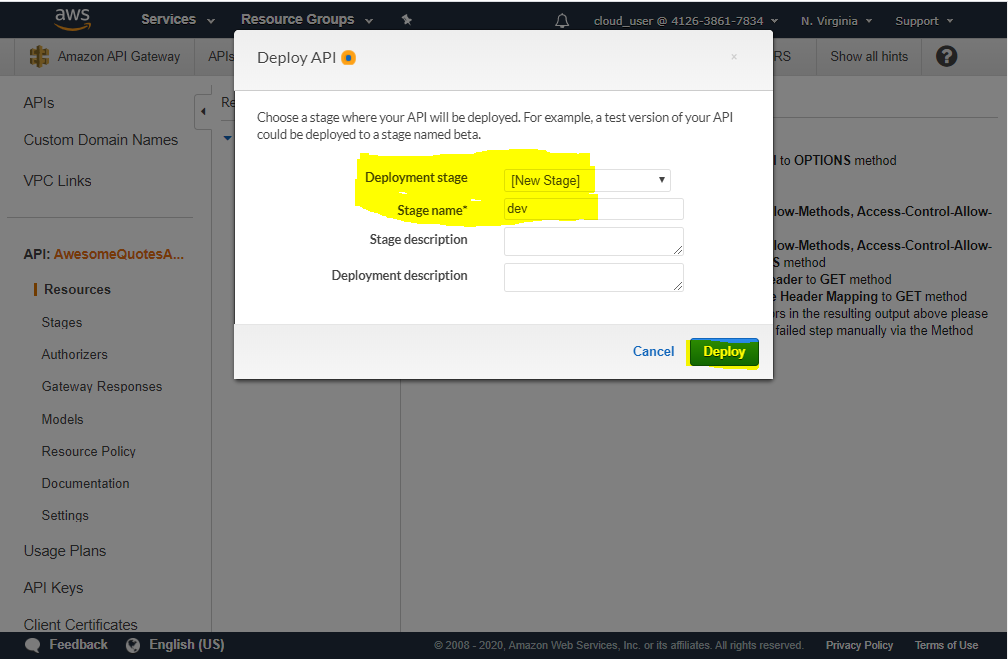




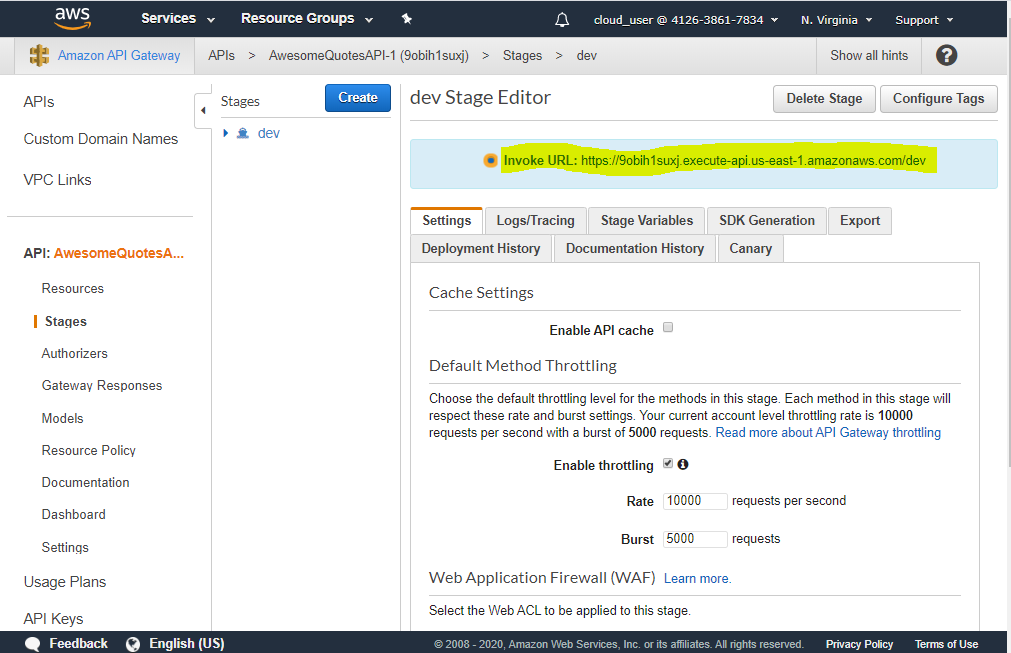
1. Next, ensure that the root path “/” is selected, go to Actions 🡪 Deploy API



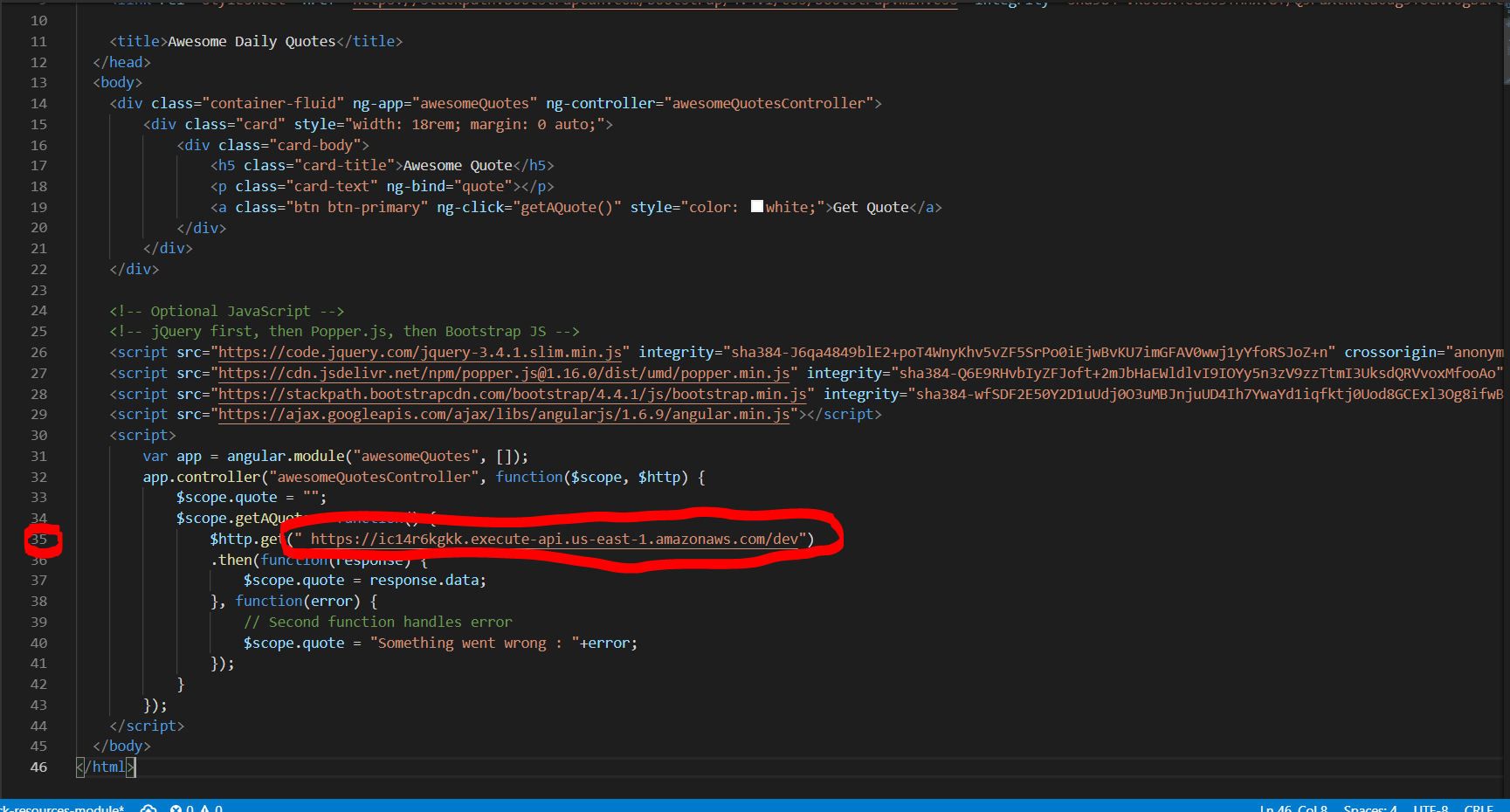
1. In the popup, select
   1. Deployment Stage : [New Stage]
   2. Stage name\* : dev
   3. Click on [Deploy] button



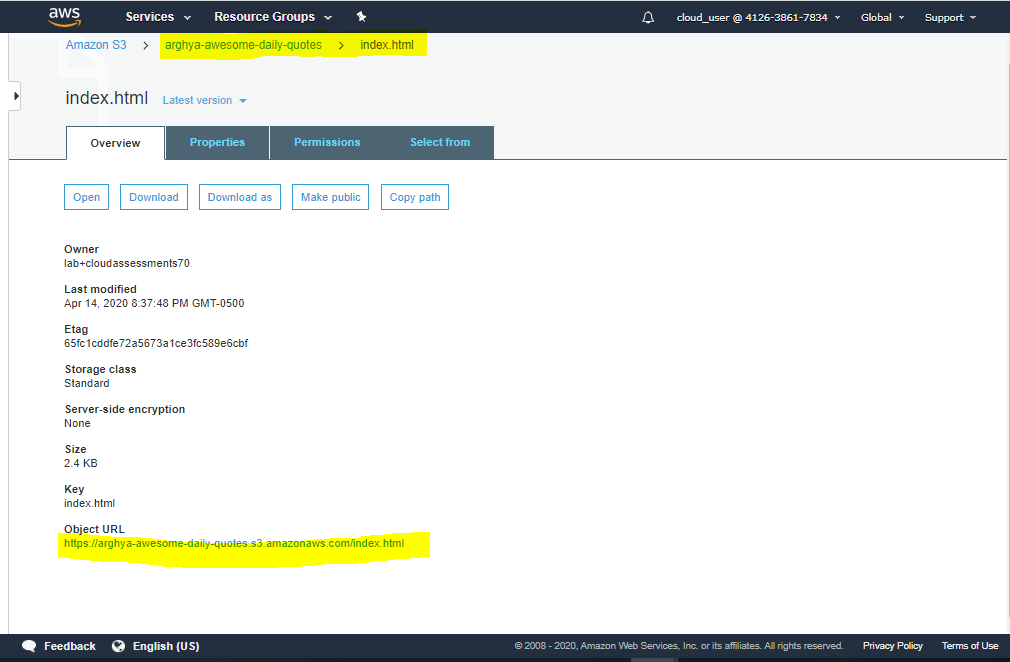
1. Ignore the red popup that appears. Copy the “Invoke URL”

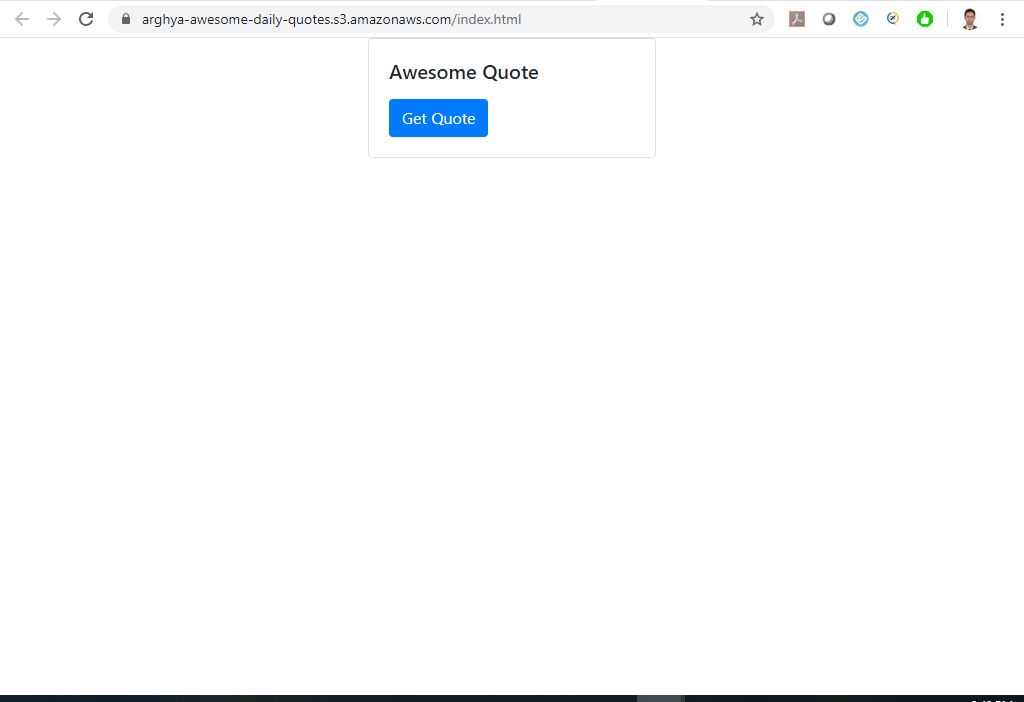


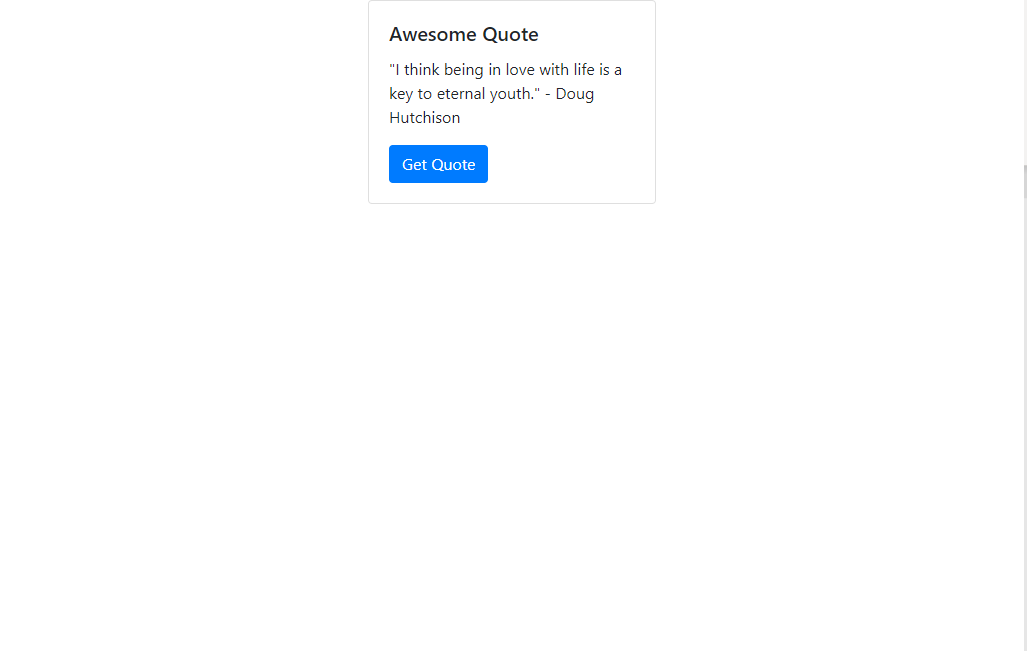
1. Now, go back to “index.html” in your local file system 🡪 change the url at line# 35 with the Invoke URL just copied 🡪 save the file 🡪 re-upload the file in corresponding S3 bucket 🡪 Once the file is uploaded, click on it 🡪 go to Permissions 🡪 Give Public Read access (see step# 21)



1. At this point, go back to <YOUR\_NAME>-awesome-daily-quotes bucket 🡪 Click on index.html 🡪 Select the Object URL 🡪 open it in a browser window 🡪 fetch some awesome & random daily quotes !!!







1. This concludes the exercise