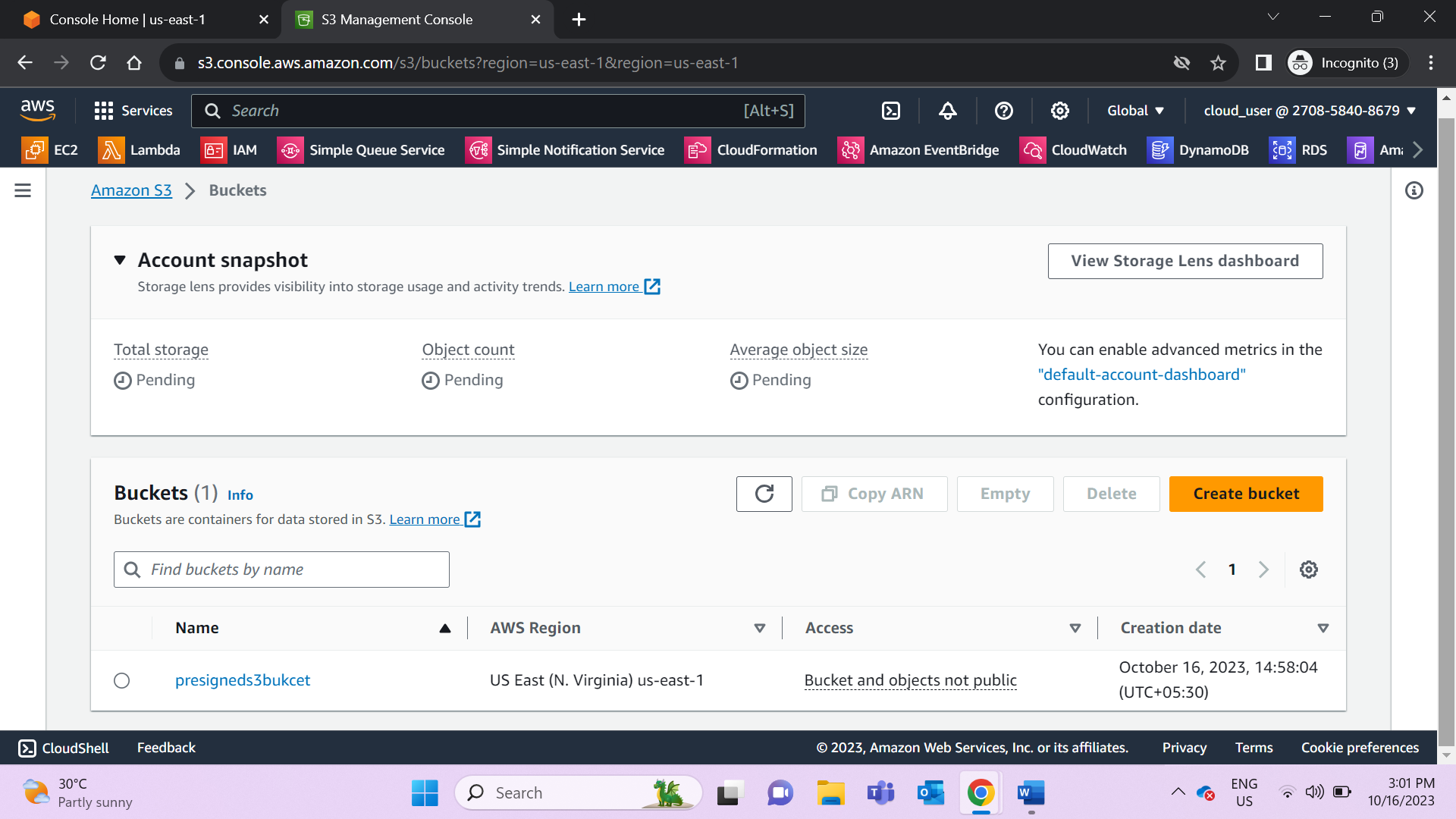
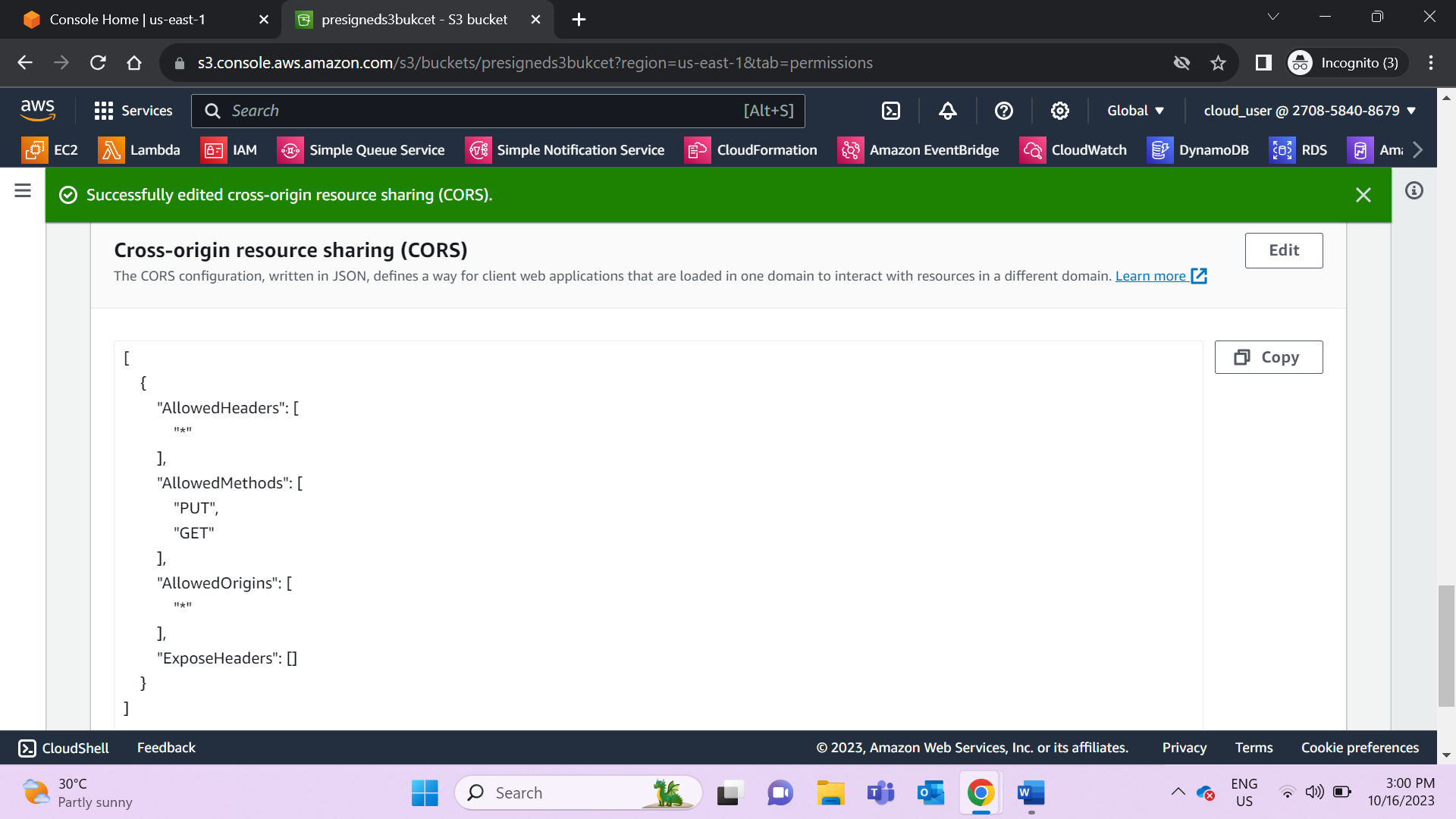
1. ***Get a pre-signed URL from our S3 bucket***
2. ***Make a request using that URL to upload your file to our bucket.***

Components involved:

Amazon S3, AWS Lambda, Amazon API GateWay and Application

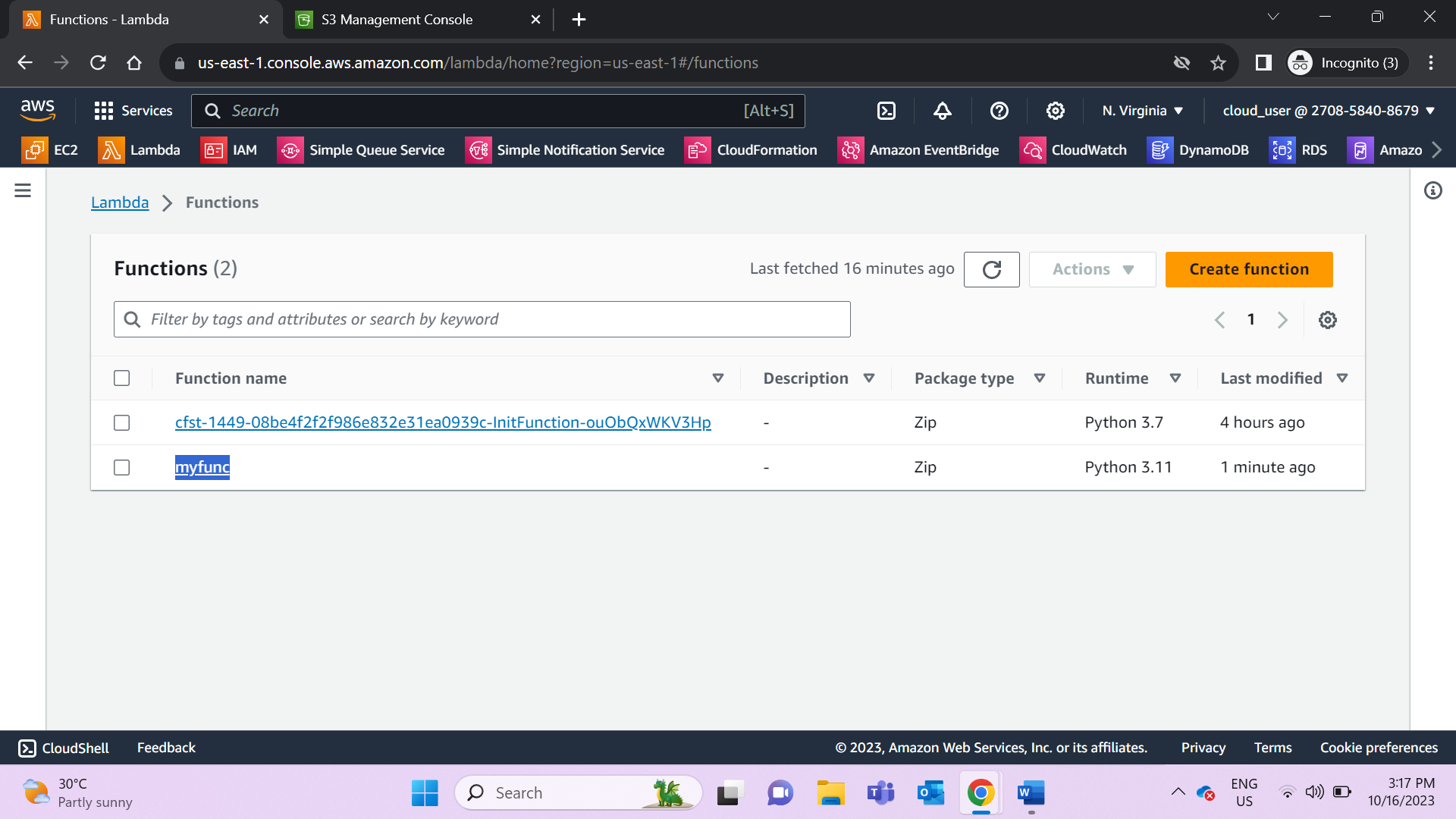
1. **Setting up Amazon S3**1.1 Create an S3 bucket

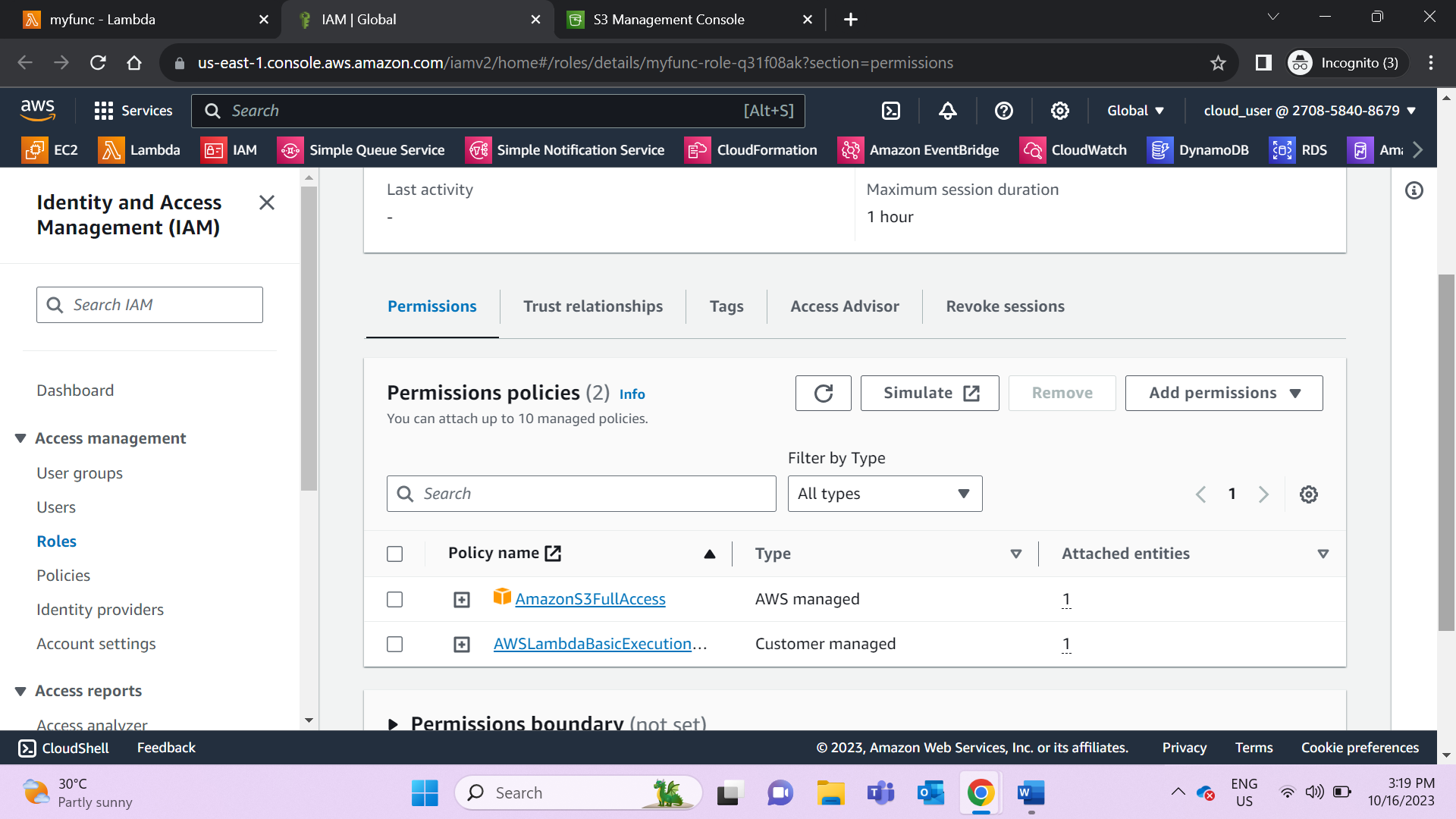
  
1.2 Enable CORS for S3 bucket



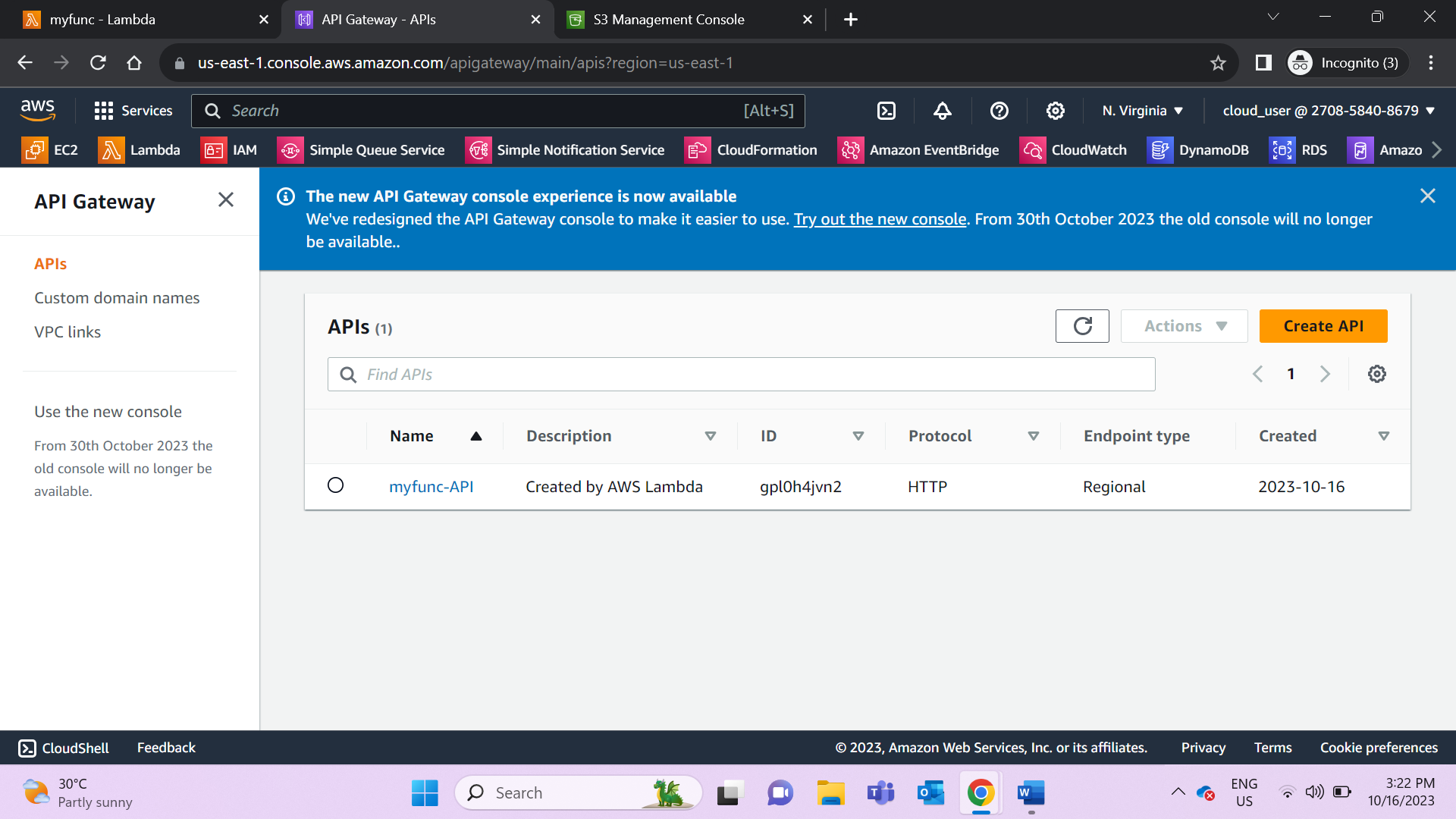
[  
{  
"AllowedHeaders": [  
"\*"  
],  
"AllowedMethods": [  
"PUT",  
"GET"  
],  
"AllowedOrigins": [  
"\*"  
],  
"ExposeHeaders": []  
}  
]

1. **Setting up AWS Lambda**2.1 Create an AWS Lambda function  
   2.2 Test the Lambda function

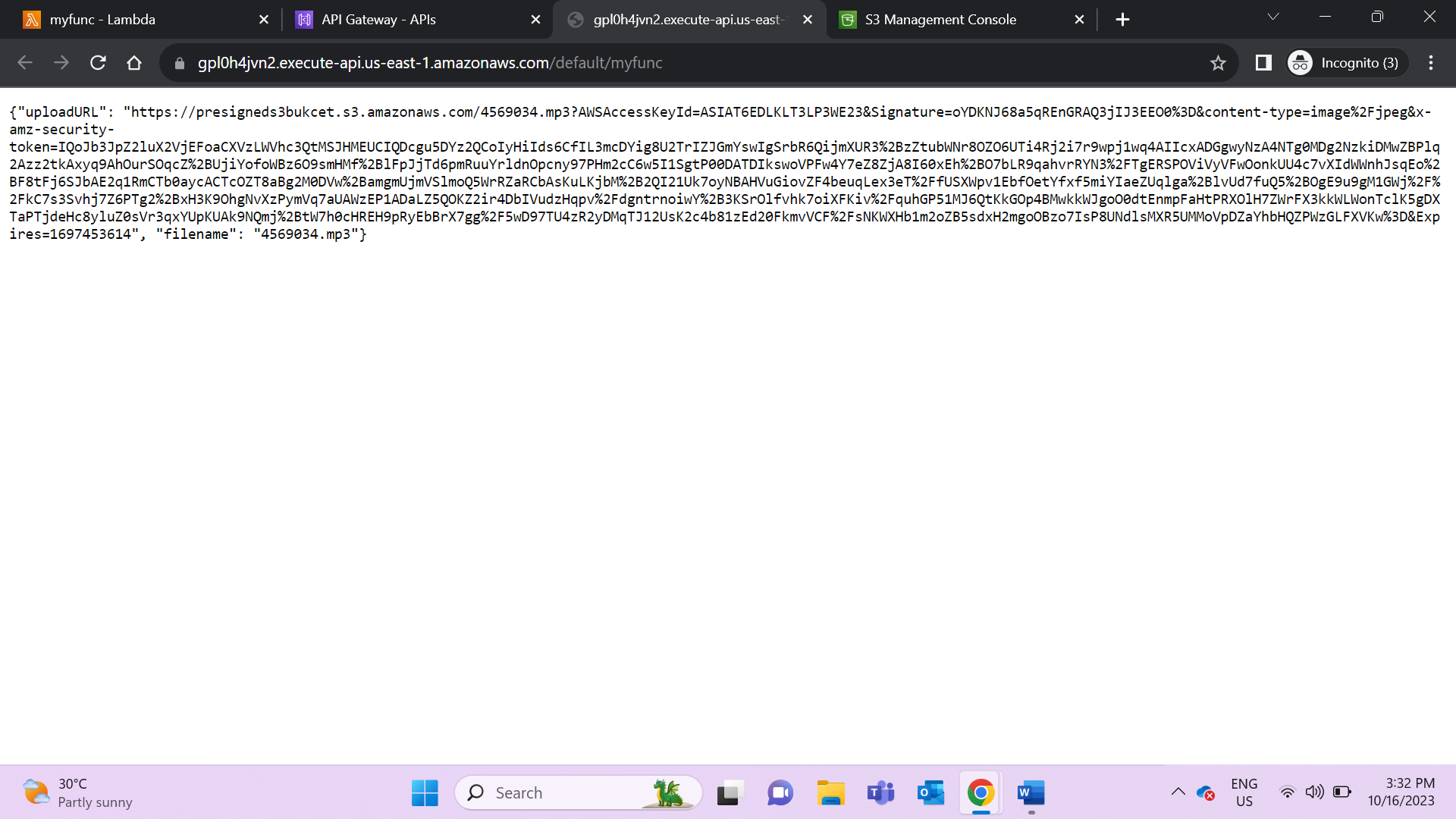
  
2.3 Enable S3 access for the Lambda function IAM role



1. **Setting up Amazon API GateWay**3.1 Create an HTTP API Endpoint; Set security to “Open” and turn on CORS:  
   3.2 Test the endpoint using Postman

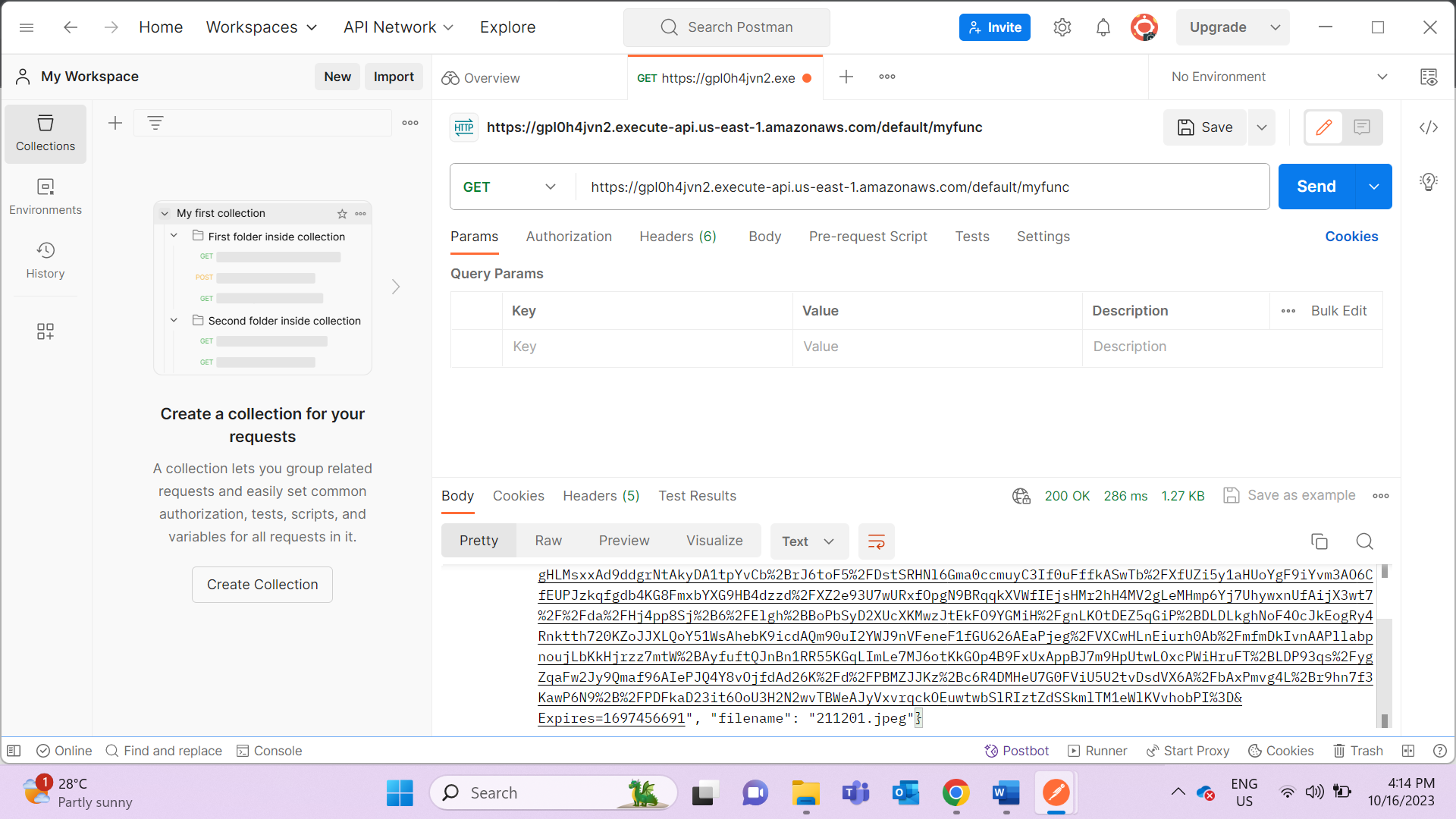


Pre-signed URL generated

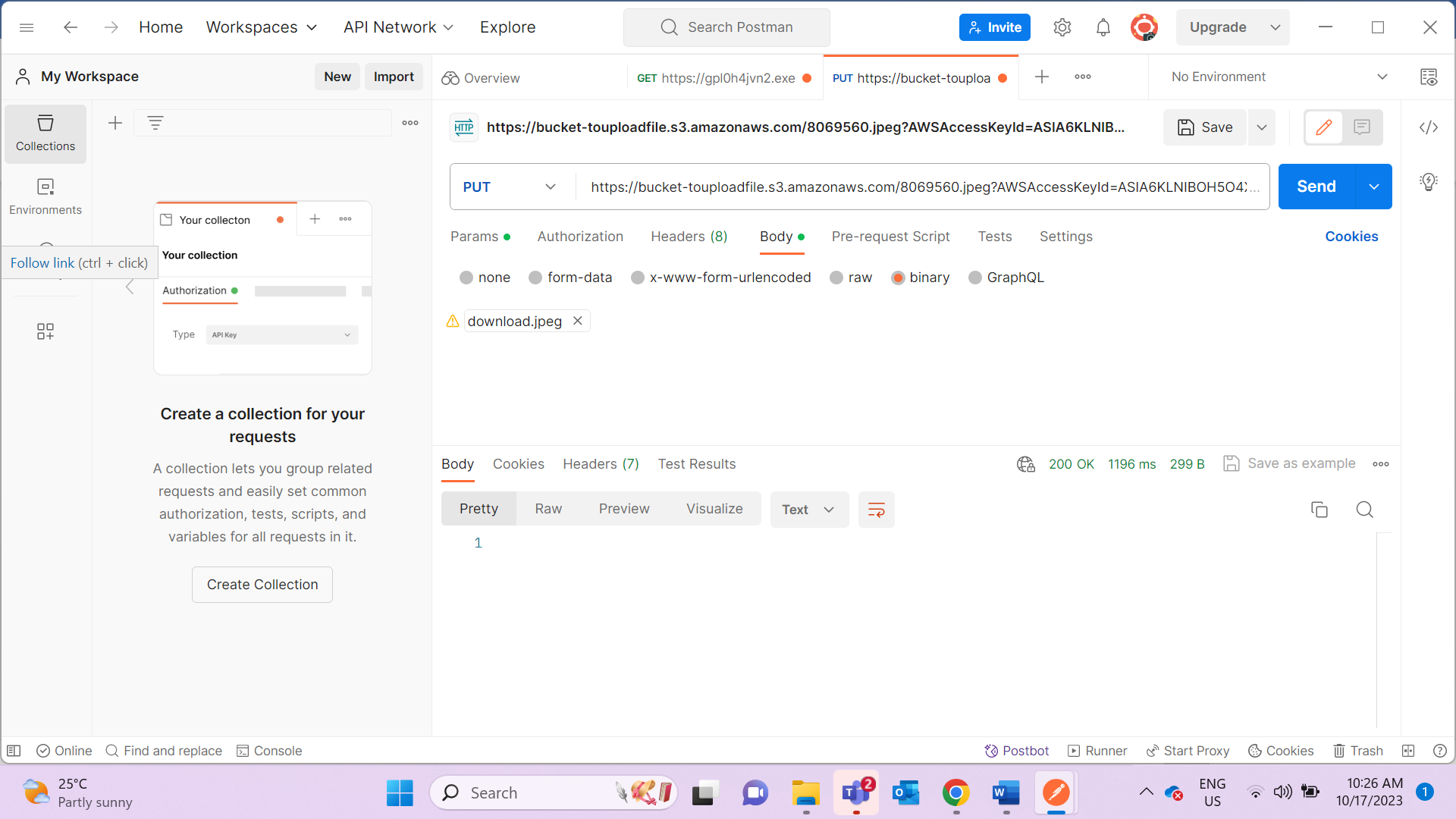


Pre-signed URL generated using Postman

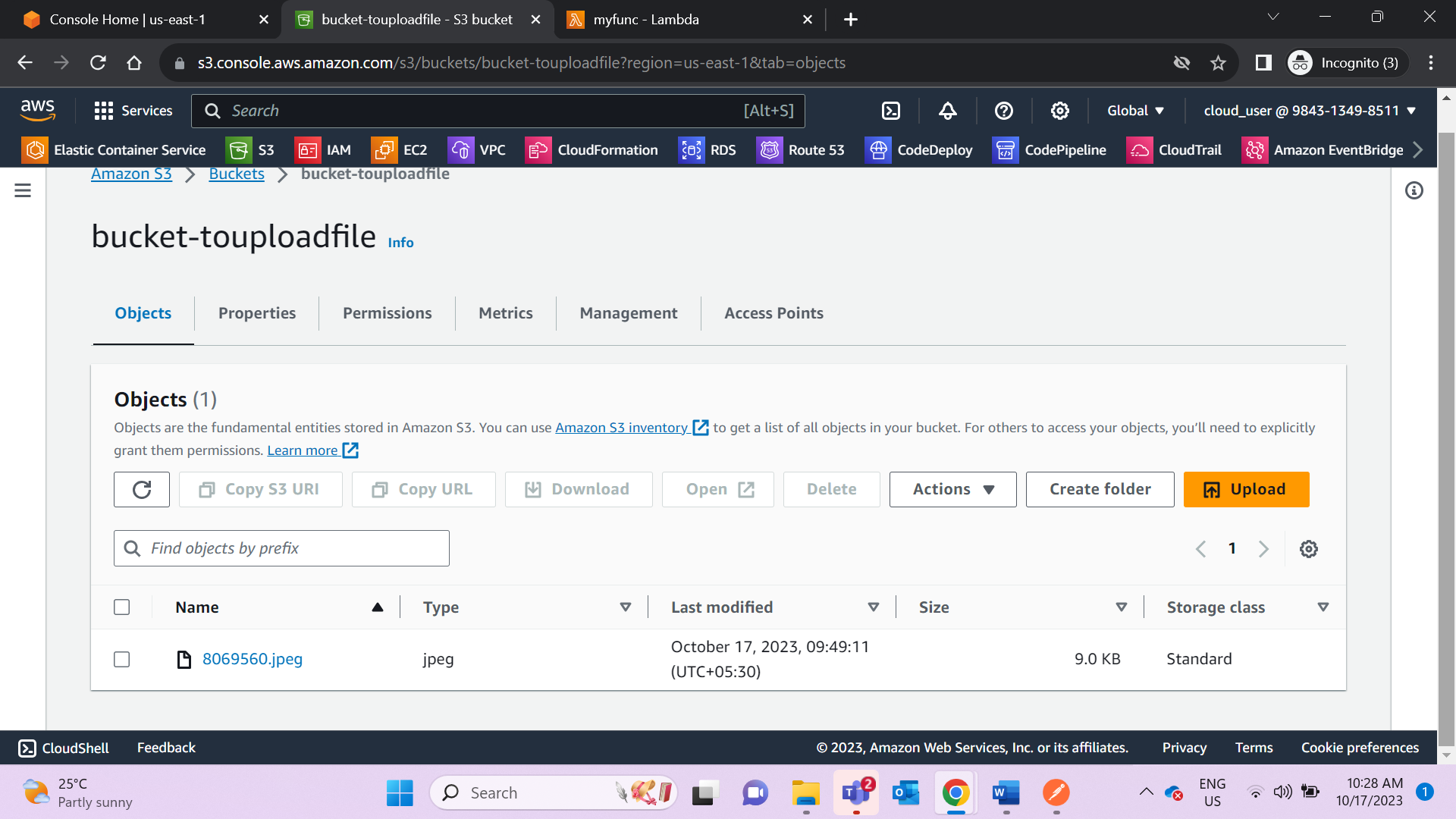
Copy the API endpoint and paste it on Postman with Get command :



Next, copy the uploadURL value and make another request to it. This time however, change the method to “PUT” and upload a sample .jpeg image to test it. To upload an image, select the ‘Body’ tab and select the ‘binary’ option which will allow you to select a file from your file system. Once done, click “Send” and if you receive no error message in response, THAT’S A GOOD SIGN.



Now, you may check your S3 bucket and you should see the file uploaded with a randomly generated filename:



Ref: <https://allardqjy.medium.com/using-pre-signed-urls-to-upload-files-to-amazon-s3-from-reactjs-5b15c94b66df>