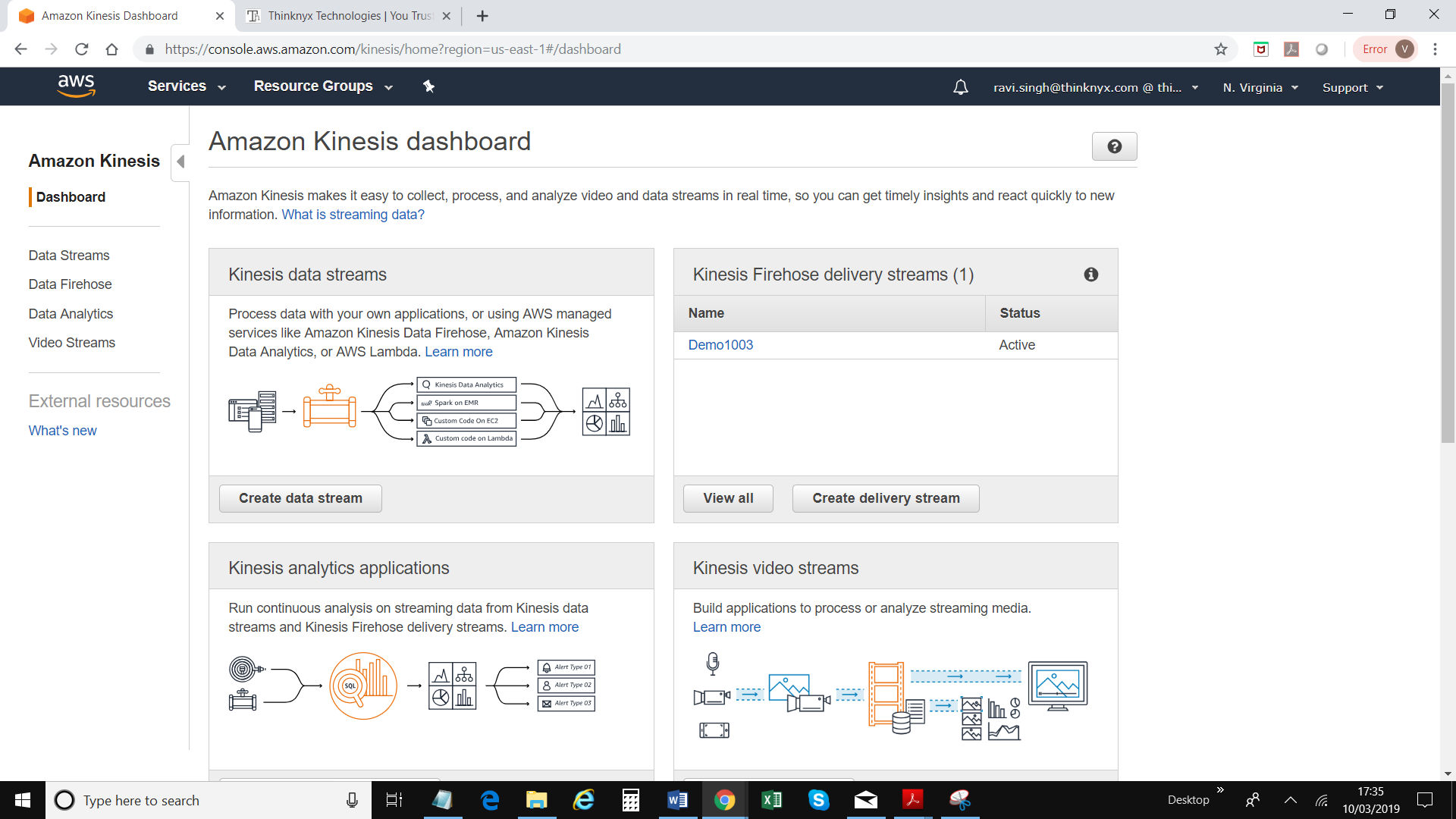
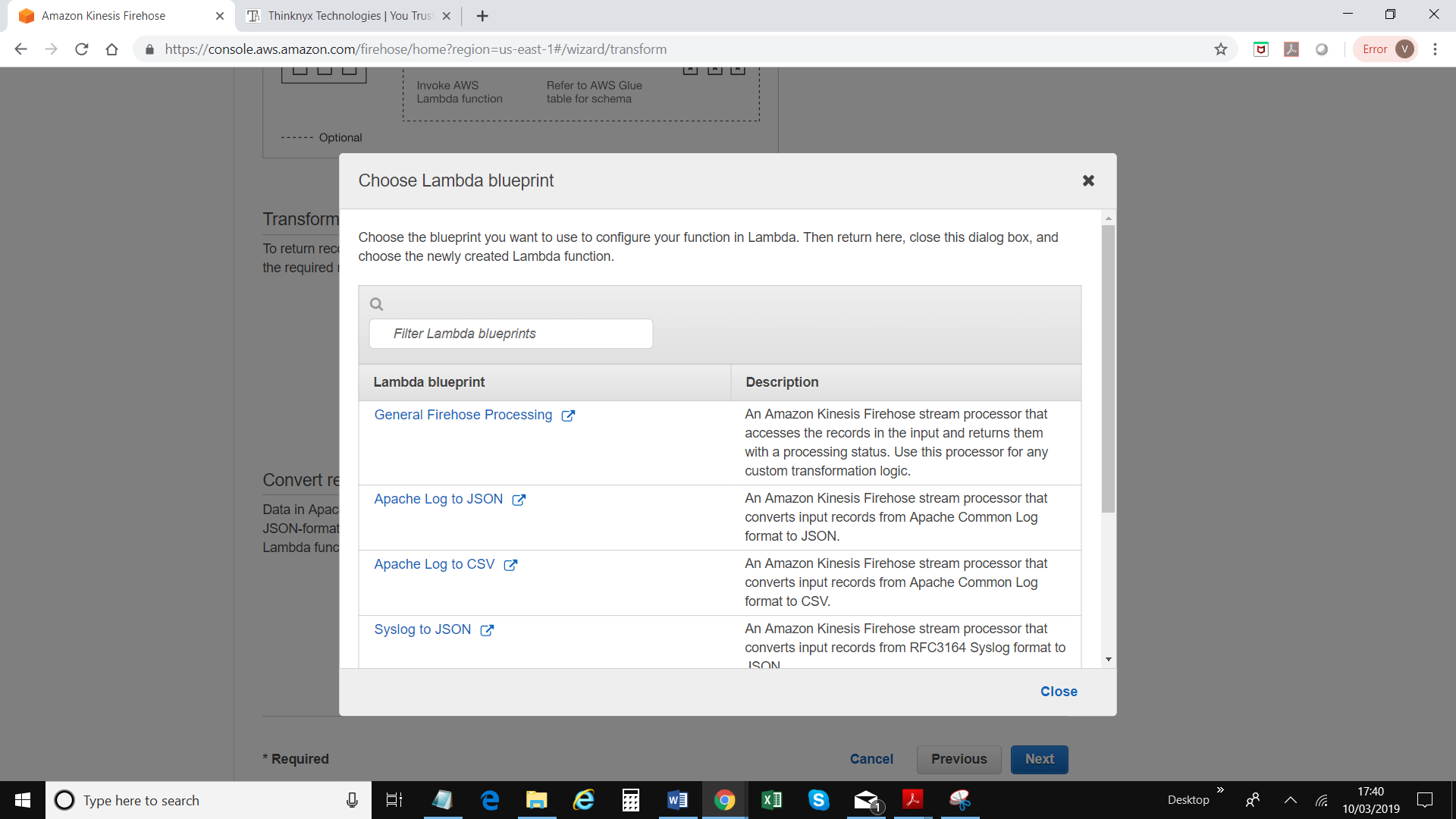
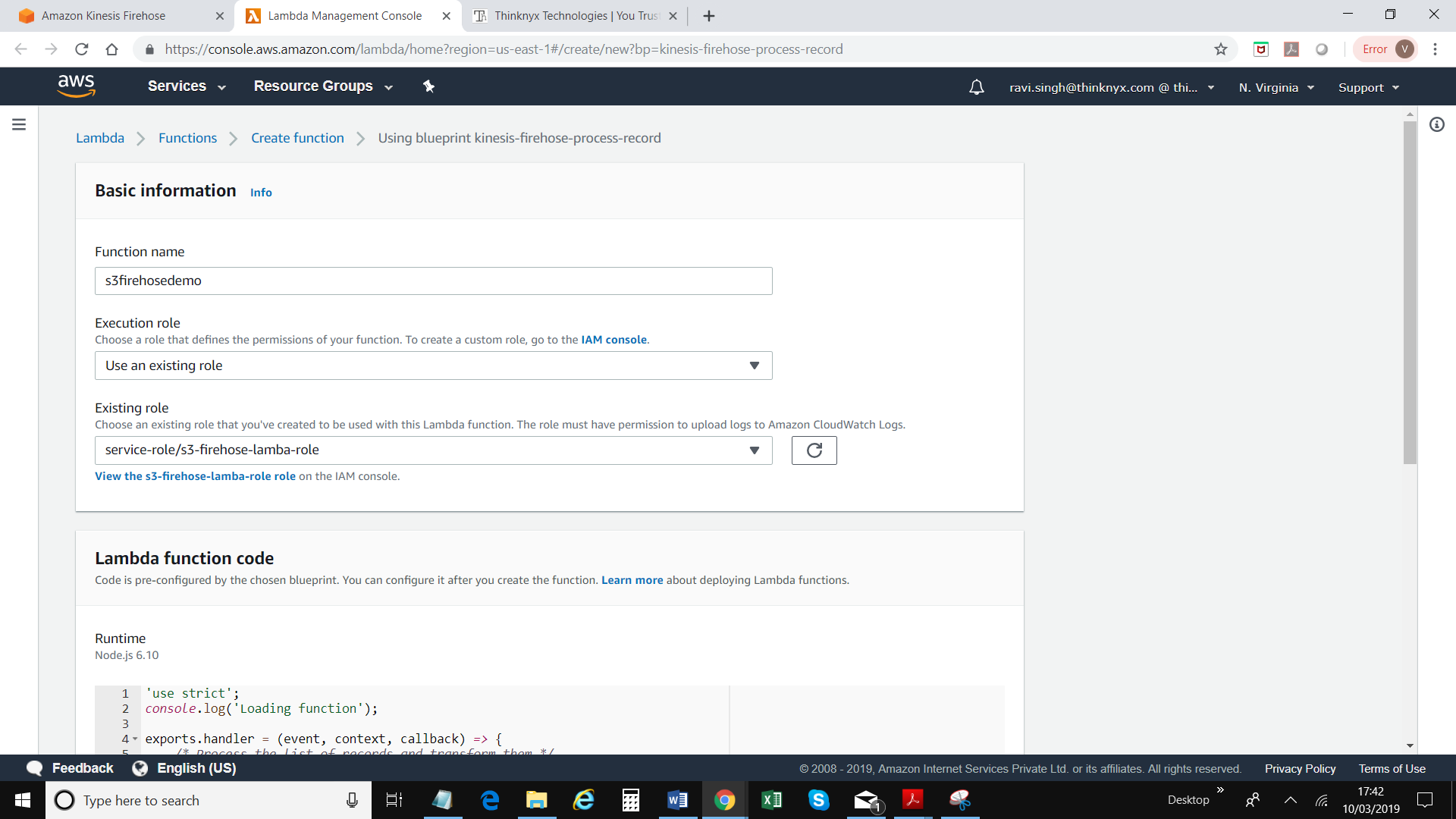
1. Go to services. Search for Kinesis



1. Click on “Create Delivery stream” under “Kinesis Firehose delivery stream”
2. Enter a “Delivery Stream Name”
3. Click “Next” – bottom right part of the screen
4. Enable “Record Transformation” & Click on “Create New” button
5. Choose “General Firehose Processing” from the list



1. A new window will be displayed
2. Enter a function name
3. Either create a new role or select a role with name starting “service-role”



1. Click on “Create function” button on right bottom corner of the screen
2. Click on the function you’ve created
3. Copy the following code into Lambda function code editor. For demo, node.js is being used.

'use strict';

console.log('Loading function');

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

/\* Process the list of records and transform them \*/

const output = event.records.map((record) => {

console.log(record);

try{

let dataStr = new Buffer(record.data, 'base64').toString('ascii');

console.log('record ' + dataStr);

dataStr = new Buffer(dataStr + "@firehoseDemo" + "\n" ).toString('base64');

/\* This transformation is the "identity" transformation, the data is left intact \*/

return { recordId: record.recordId,

result: 'Ok',

data: dataStr,

}

}catch(ex){

console.log(ex);

return record;

}

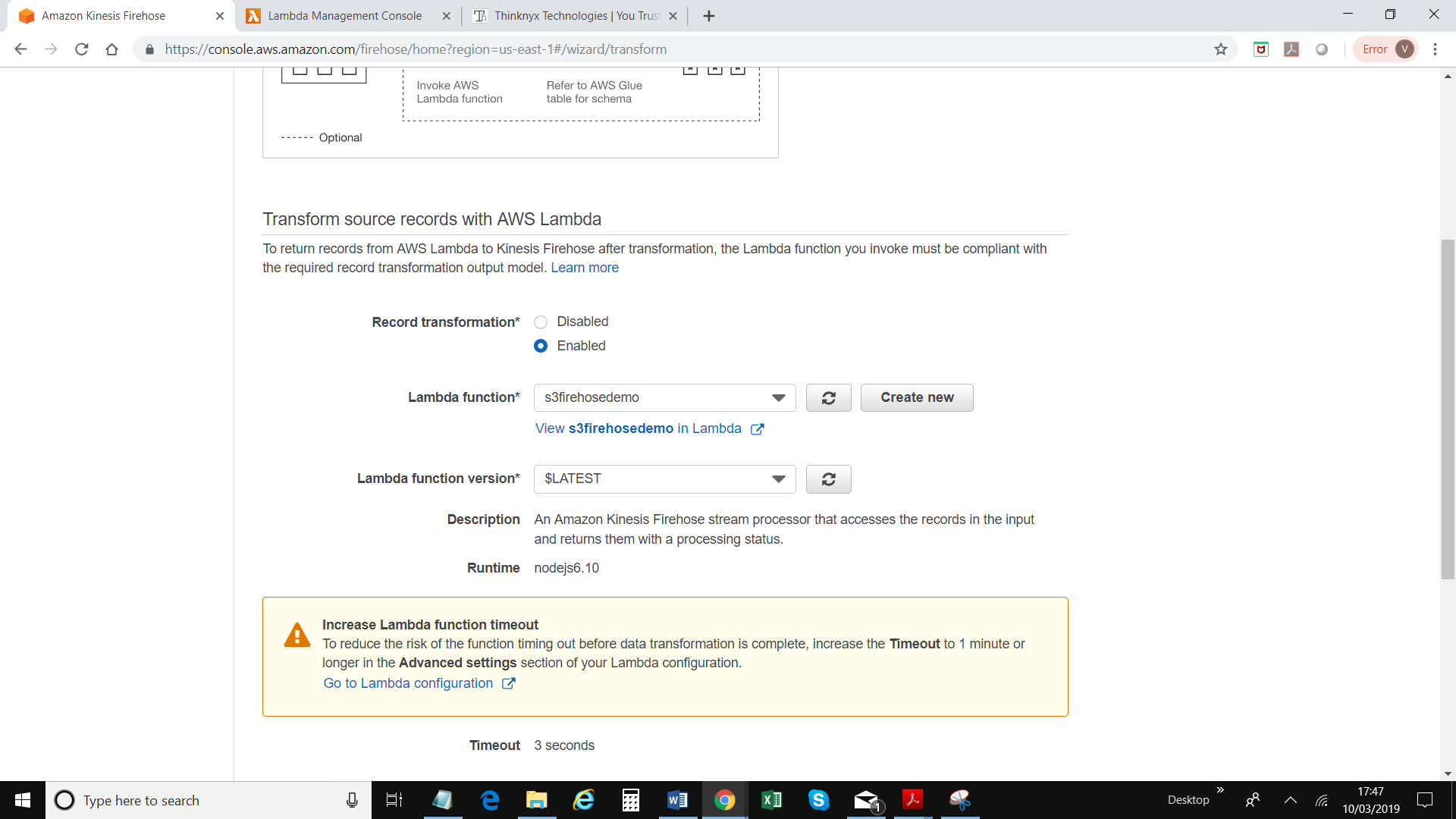
});

console.log(`Processing completed. Successful records ${output.length}.`);

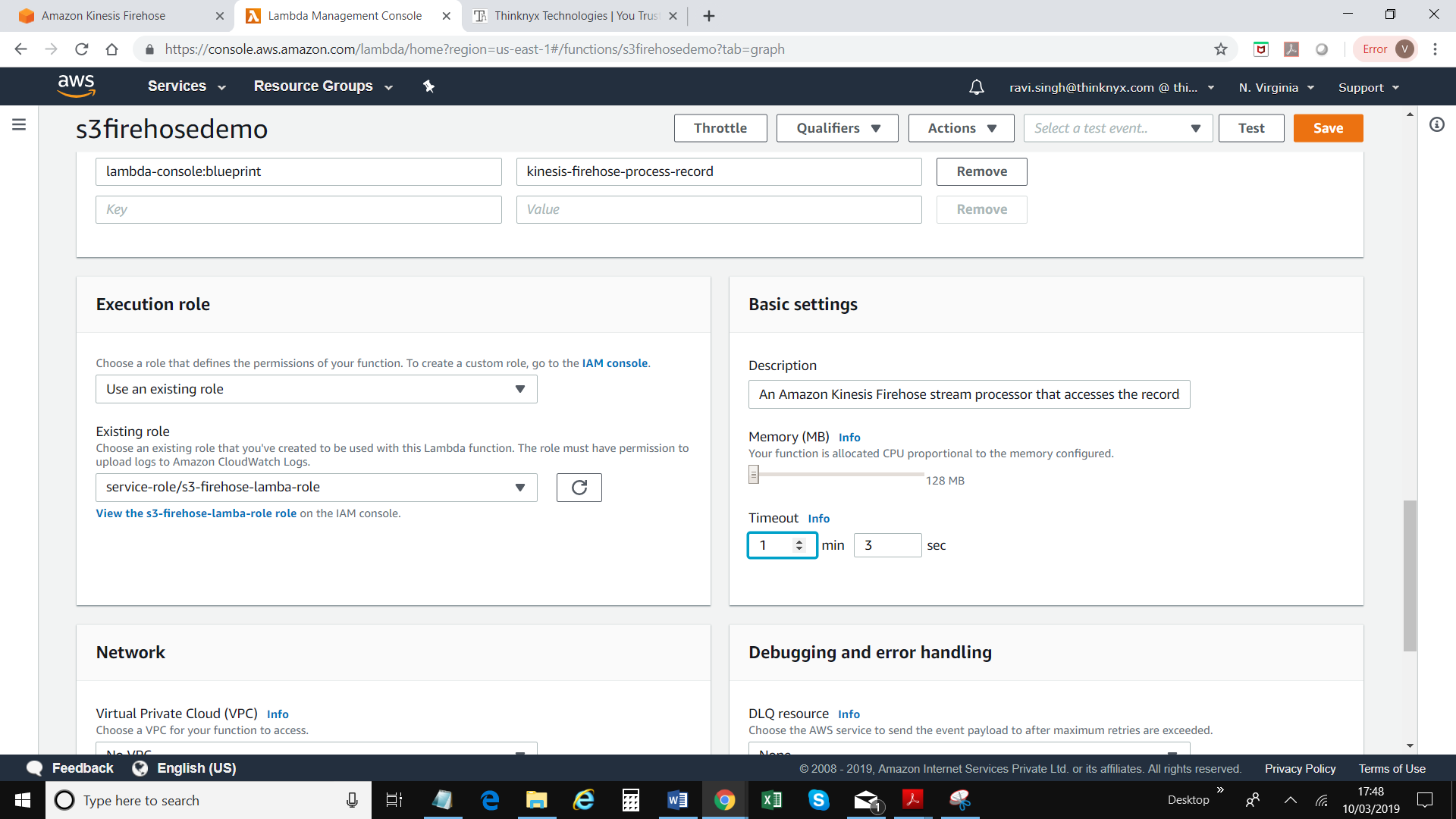
callback(null, { records: output });

};

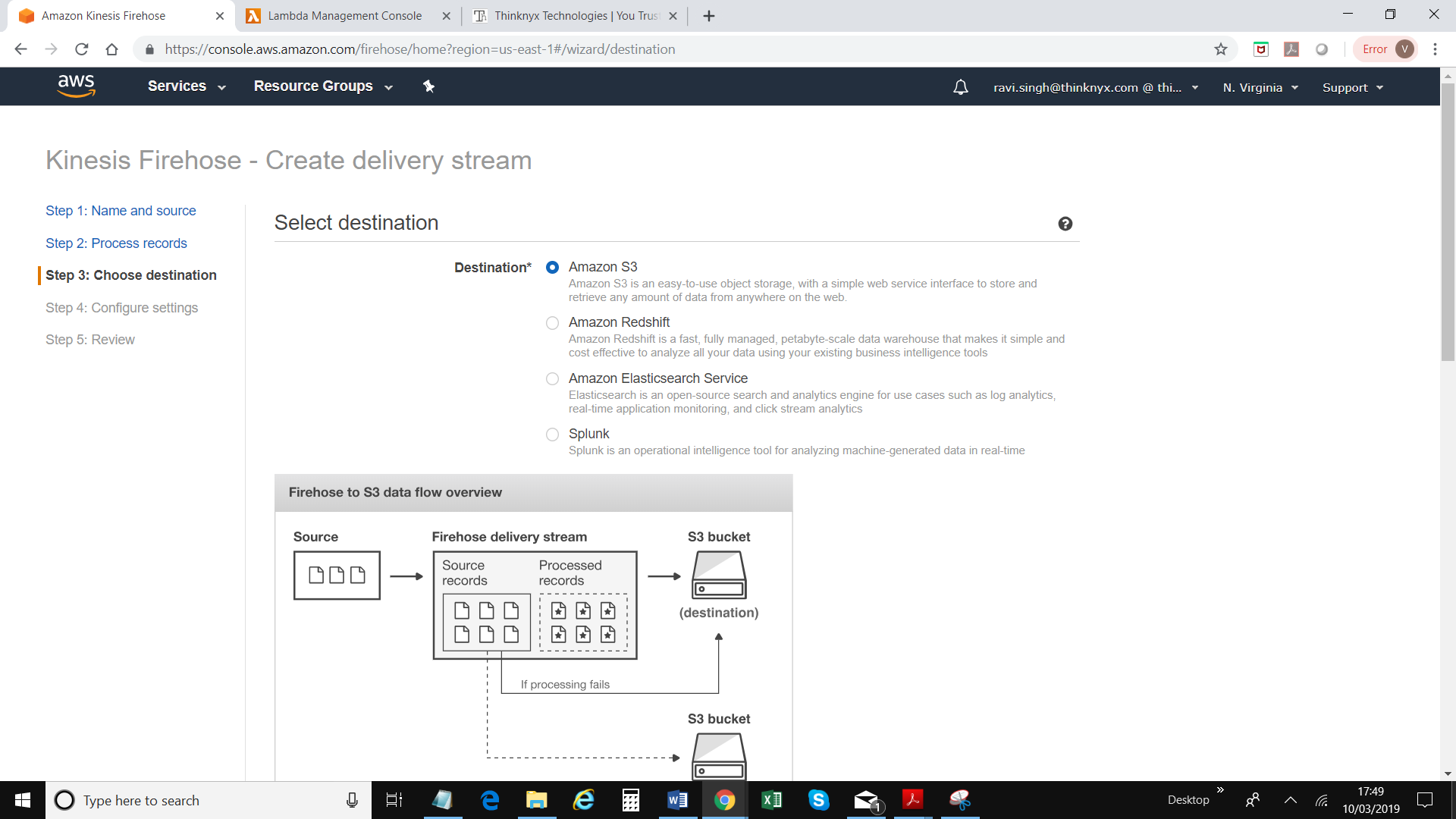
1. Click on Save button – on top of corner of the screen
2. Go back to Kinesis window / screen/ tab and close the pop up window
3. Select the Lambda function you’ve created



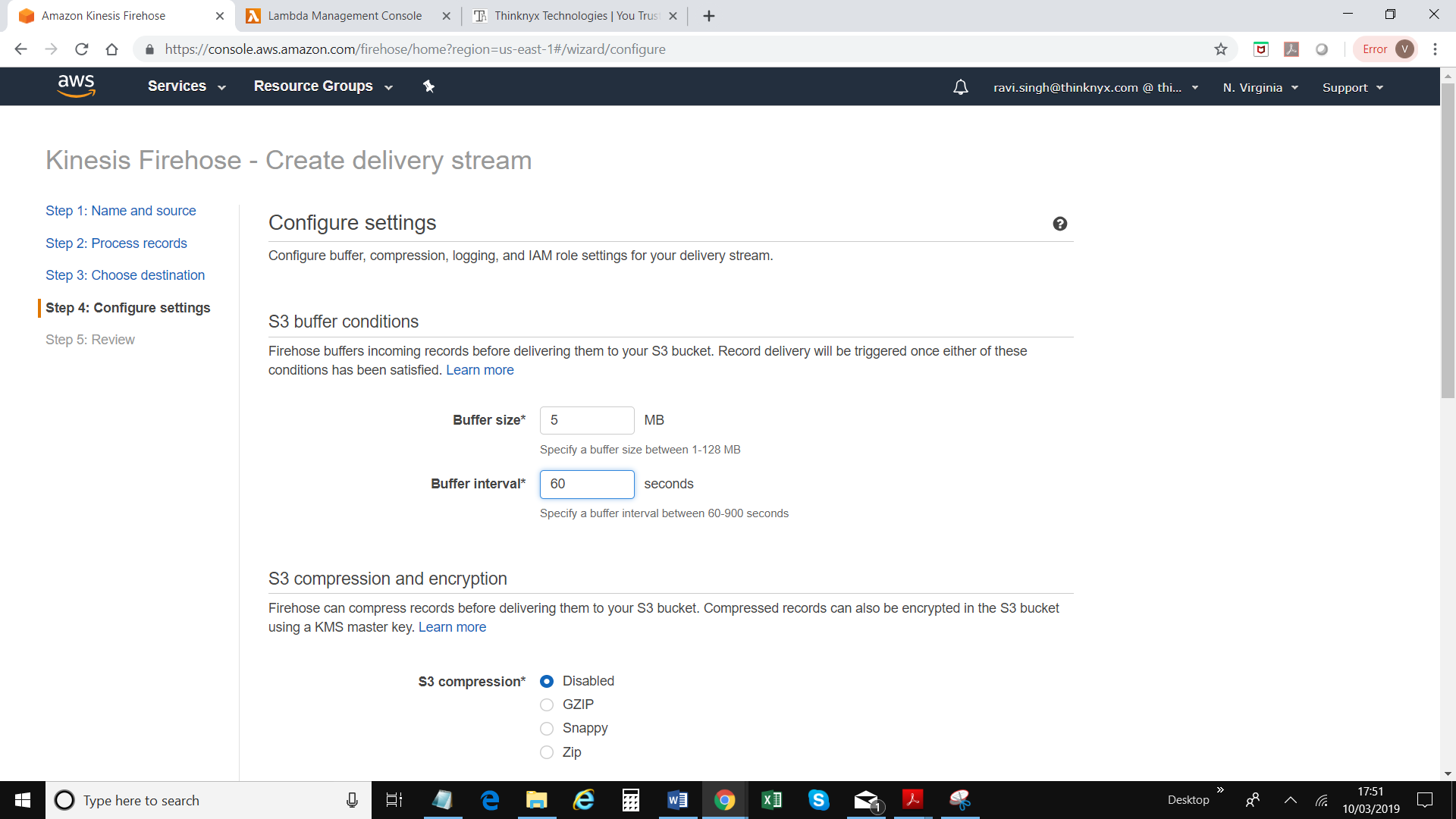
1. Go back to the lambda function you’ve created and change the timeout from 3 seconds to 1 minutes and save



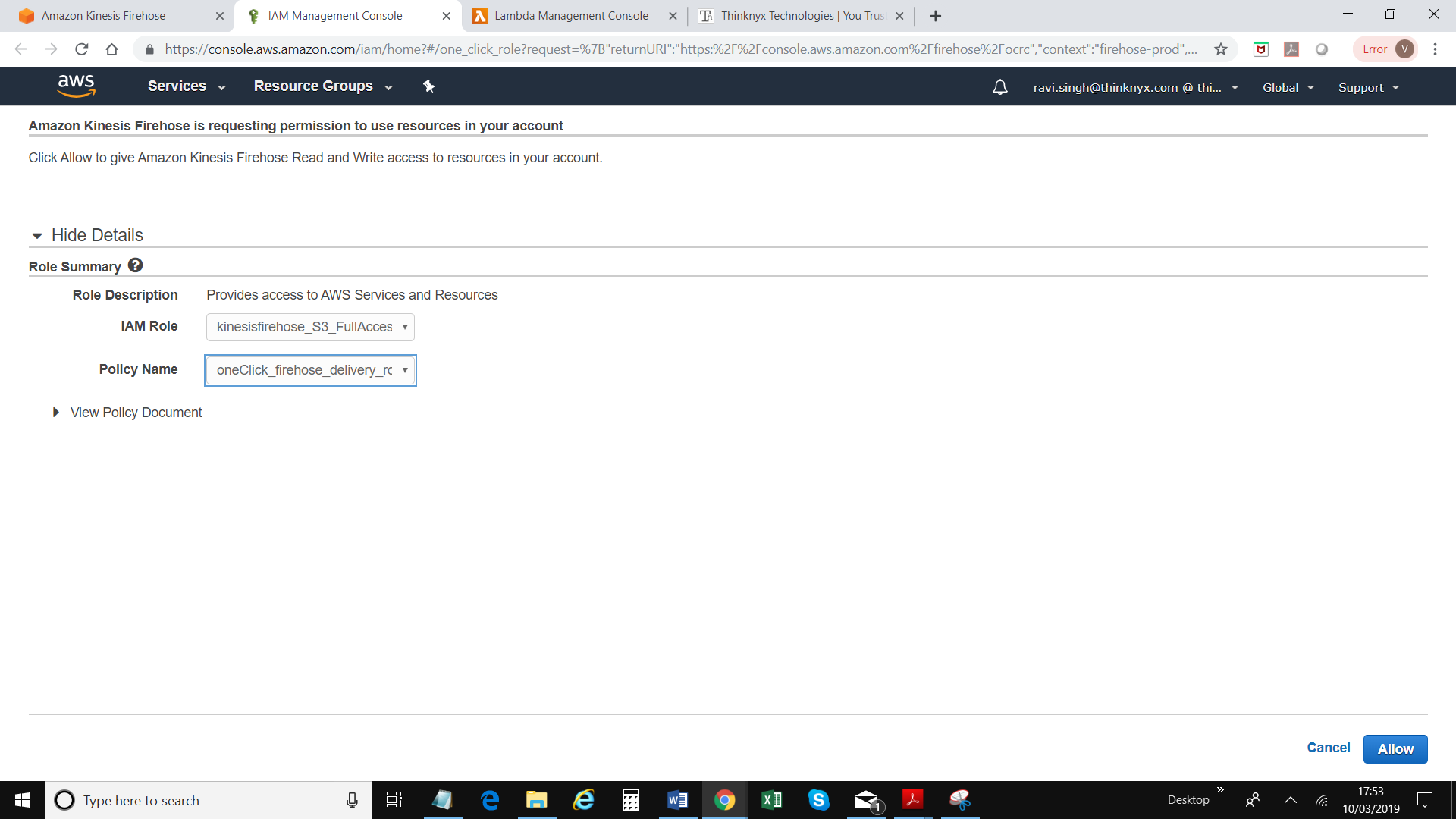
1. Back to Kinesis screen. Click on next – bottom right corner of the screen
2. There are 4 options available. For this demo, select Amazon S3



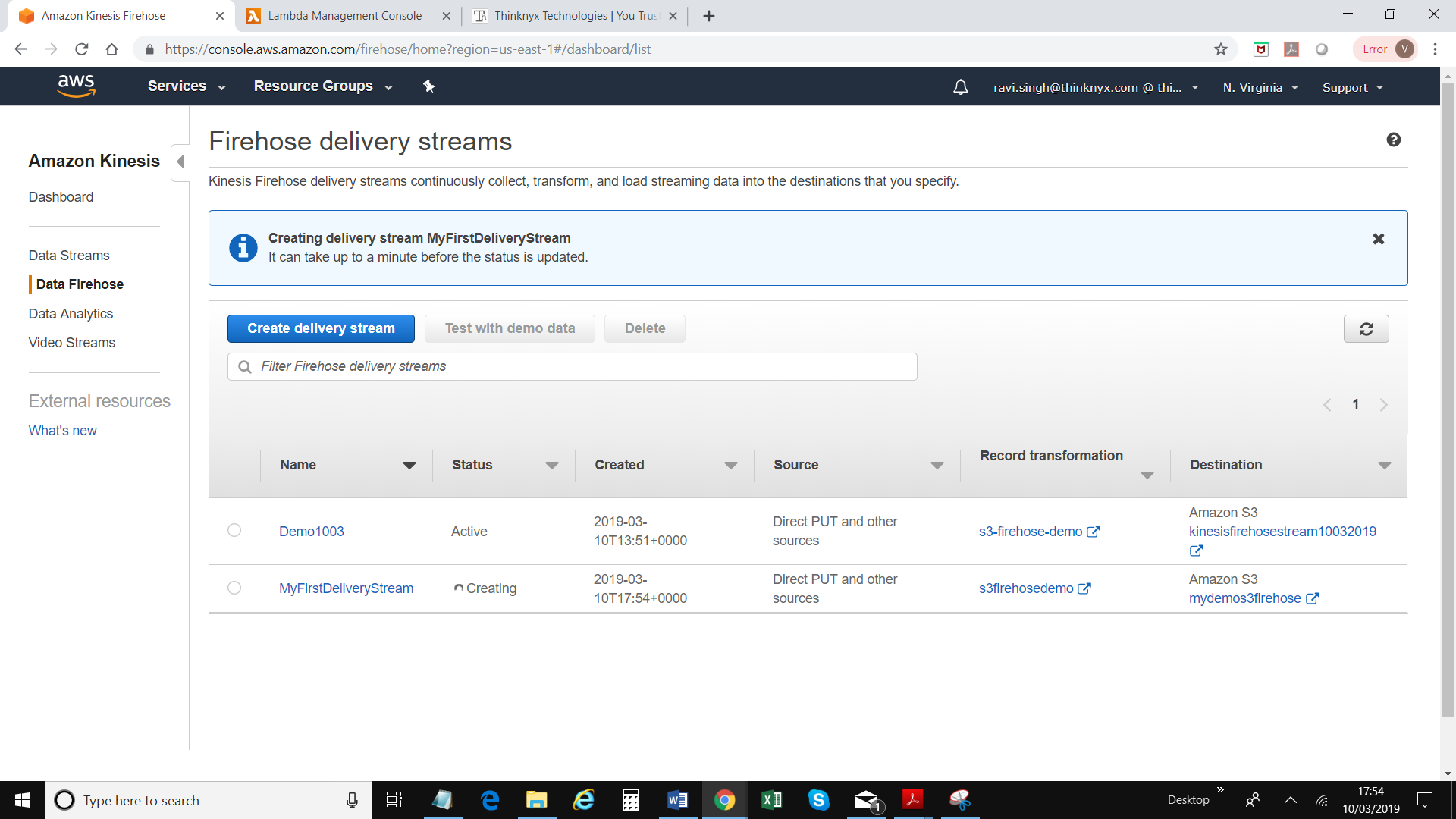
1. Select a s3 bucket from the drop down list or create a new bucket
2. Keep default values and click on “Next” – top right hand corner of the screen
3. Change buffer interval to 60



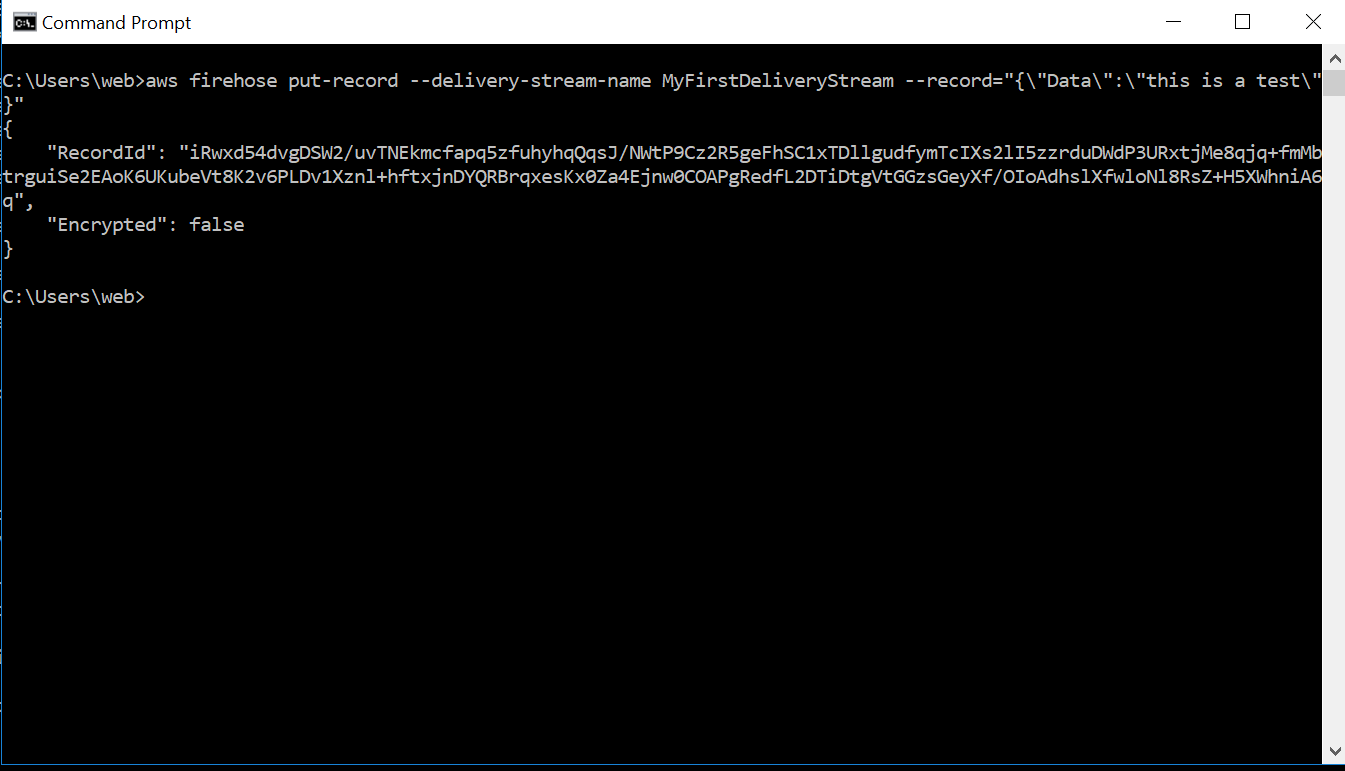
1. Create or choose existing IAM role. A new window / tab will be opened
2. Select existing IAM role and Policy Name; Click Allow



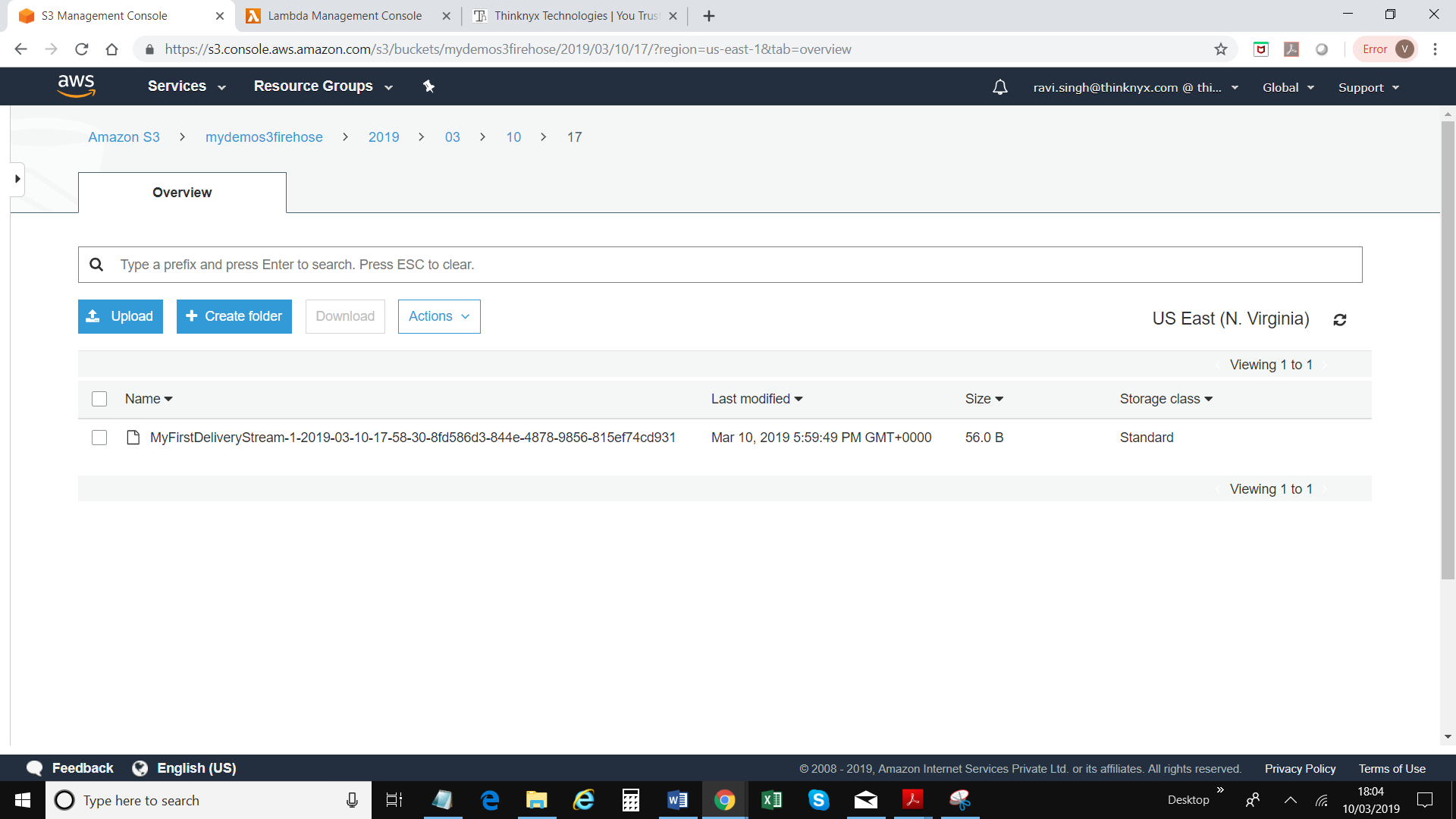
1. Click Next
2. Scroll down to the bottom of the screen, click on “Create Delivery Stream”. Following screen will be displayed



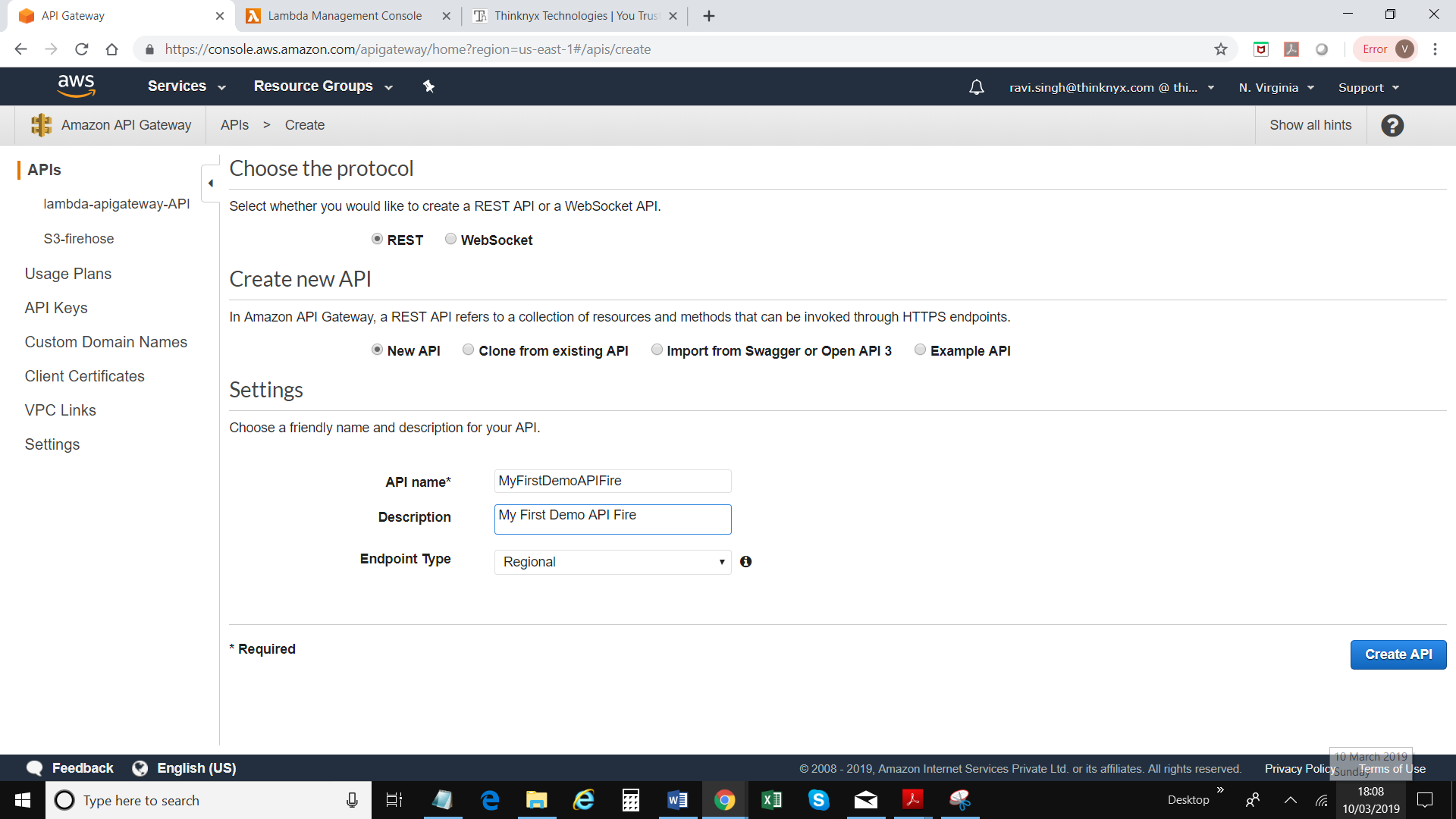
1. To send data from CLI, use - aws firehose put-record --delivery-stream-name MyFirstDeliveryStream --record="{\"Data\":\"this is a test\"}"



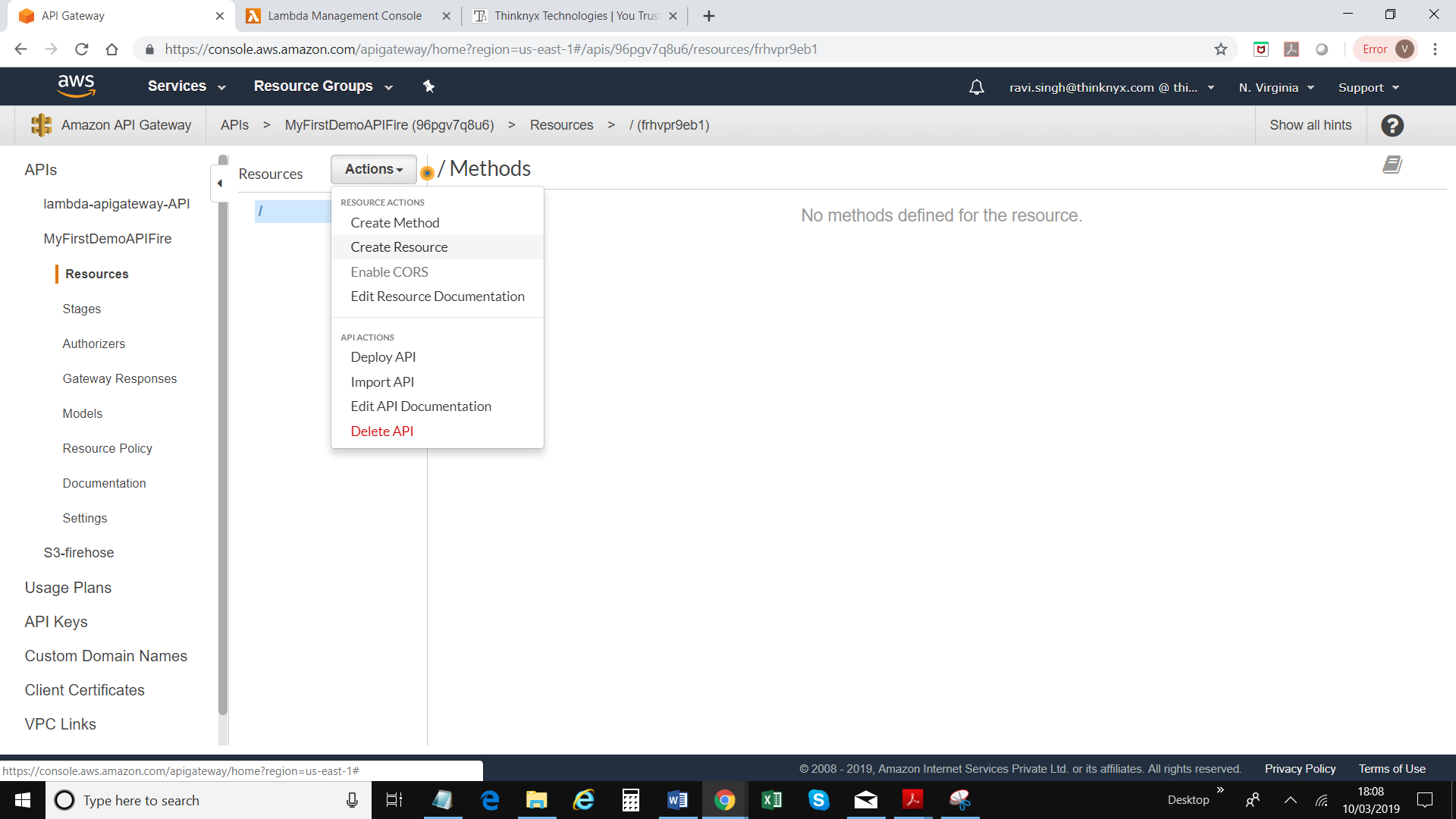
1. Go to services = > S3 and search for your S3 bucket
2. You should be able to see a record which you sent via API call



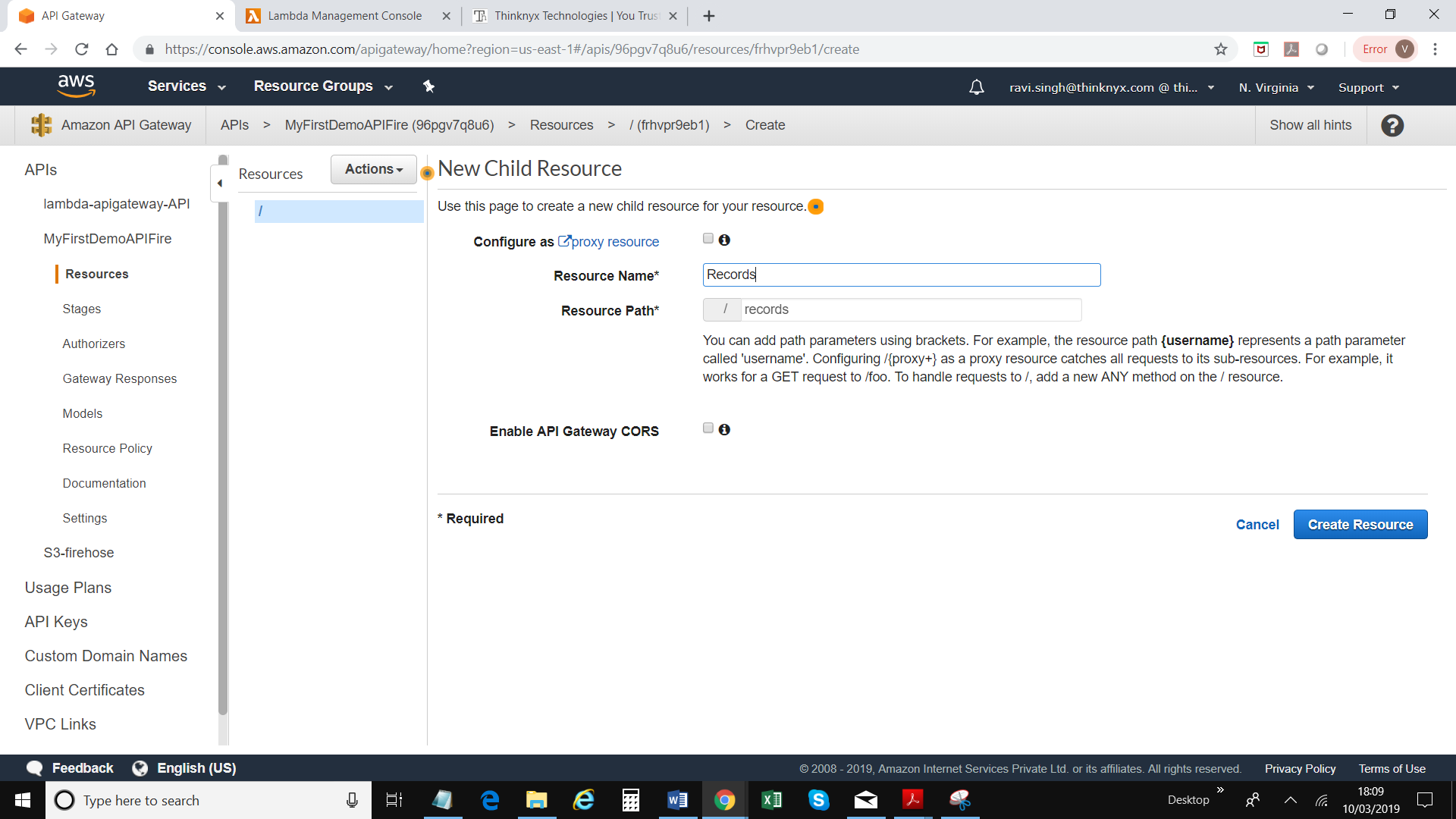
1. Click on the file and either download it and open the file using wordpad
2. Next: Expose Kinesis as an API end point via API Gateway
3. Go to Services => API Gateway
4. Click create API button to create a new API



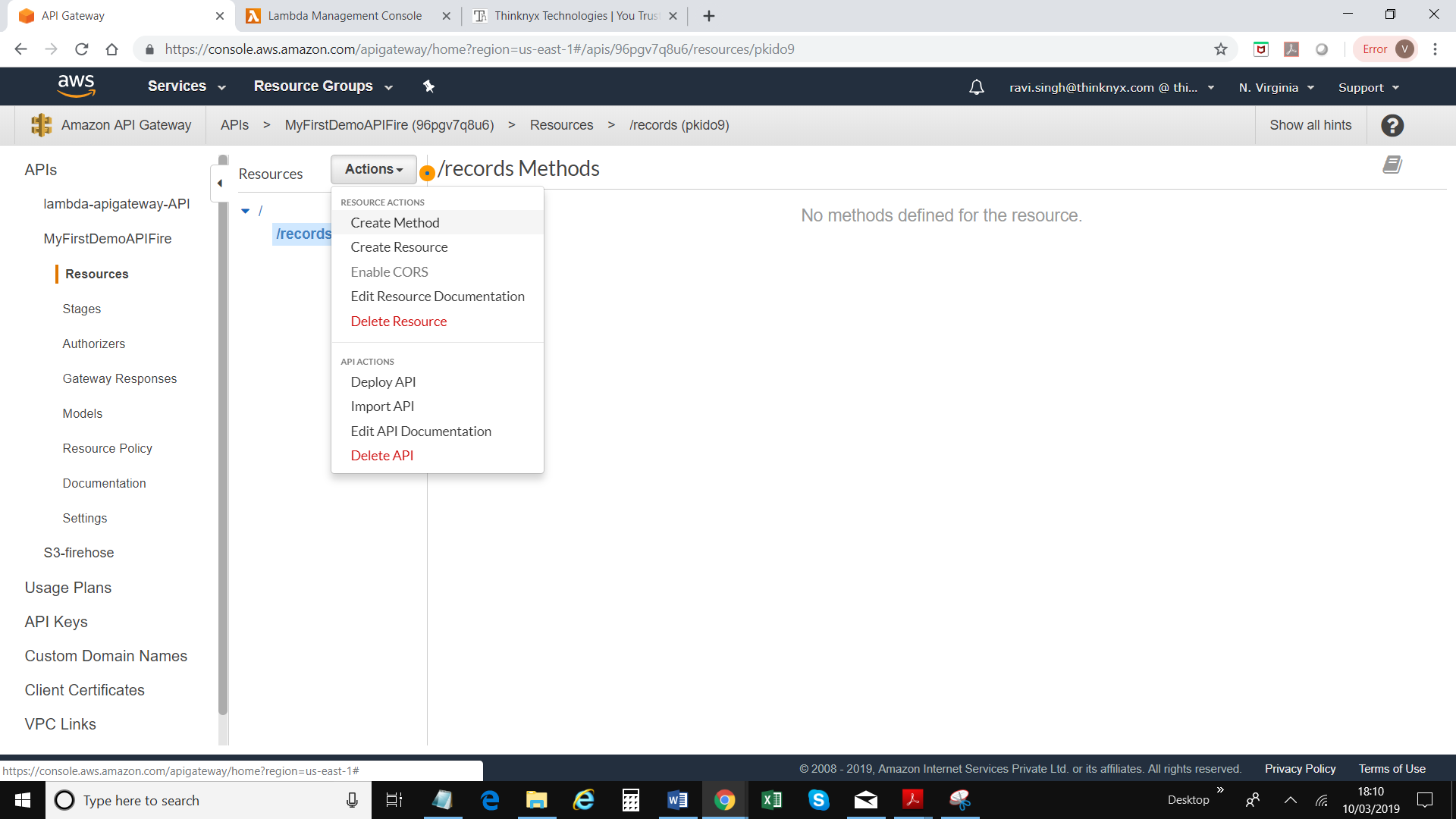
1. Click “Create API”
2. Click on Actions => Create Resource



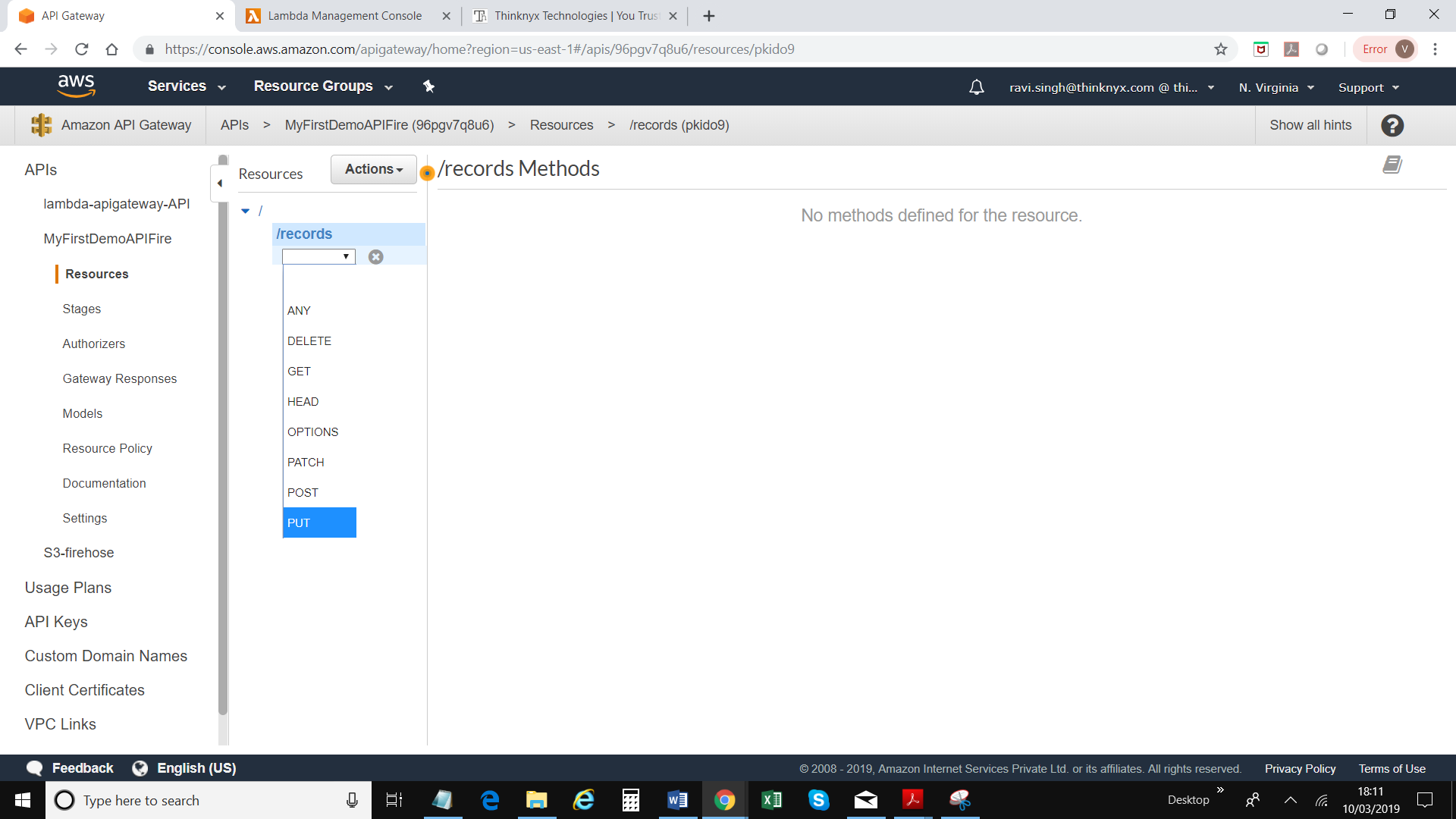
1. Type resource name and then click Create Resource



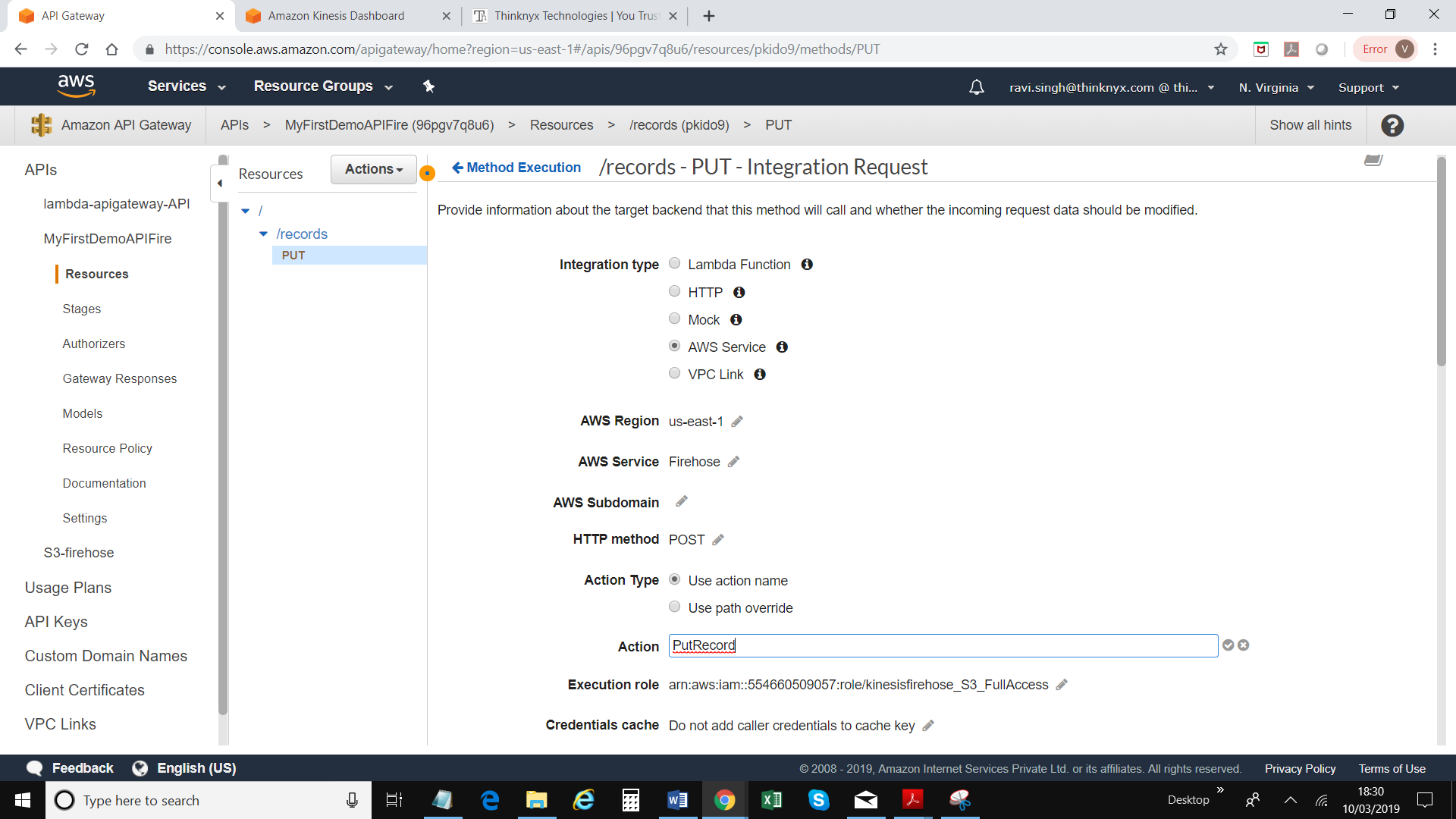
1. Now create a method by using “Create Method” under Actions



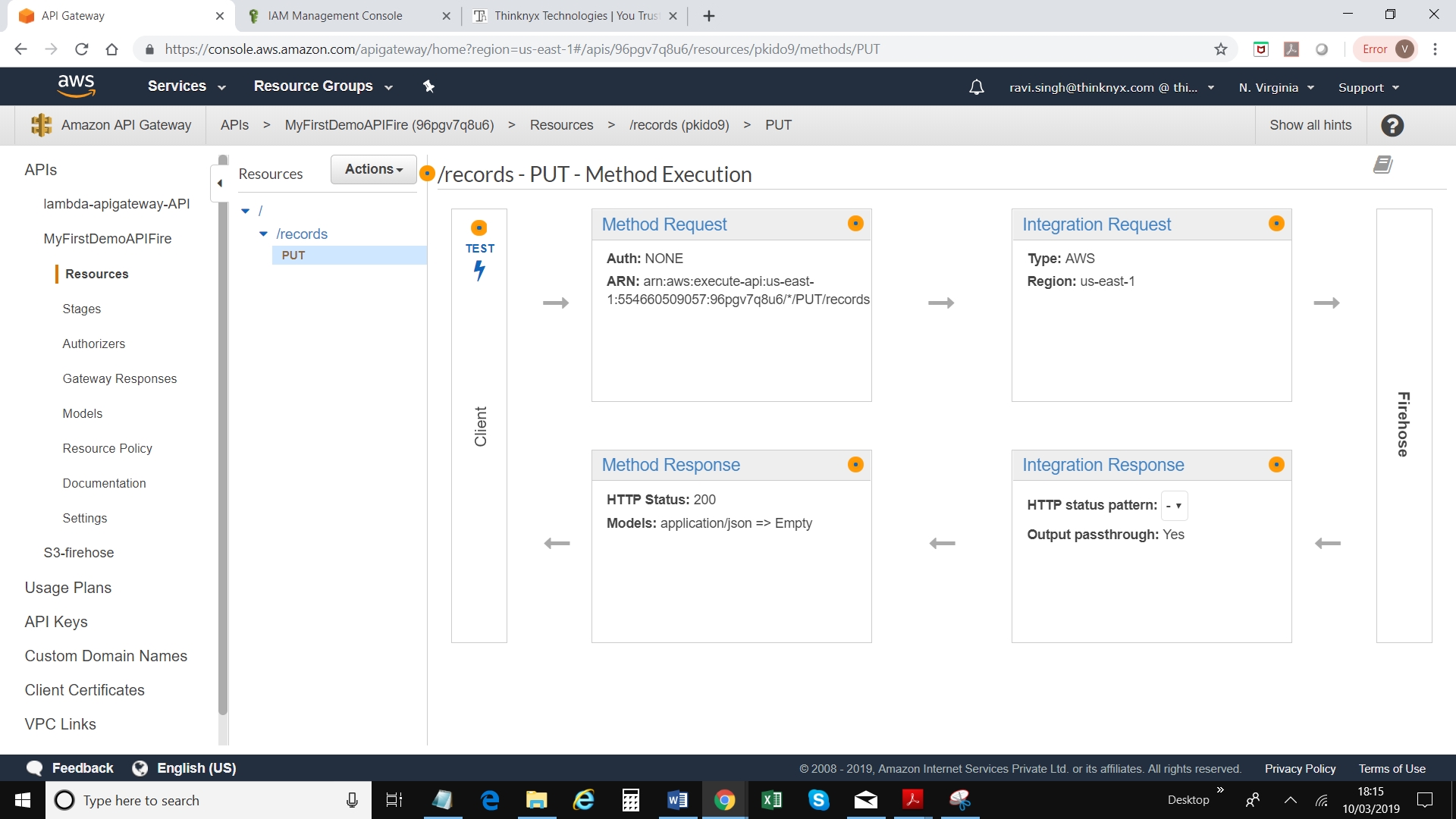
1. Choose method type Put and click right button



1. Select “AWS Service” as Integration type, regions as US-East-1, Firehose from AWS Service drop down, Post from HTTP Method, and put value arn:aws:iam::554660509057:role/kinesisfirehose\_S3\_FullAccess under “Execution role”; Action name=”PutRecord”

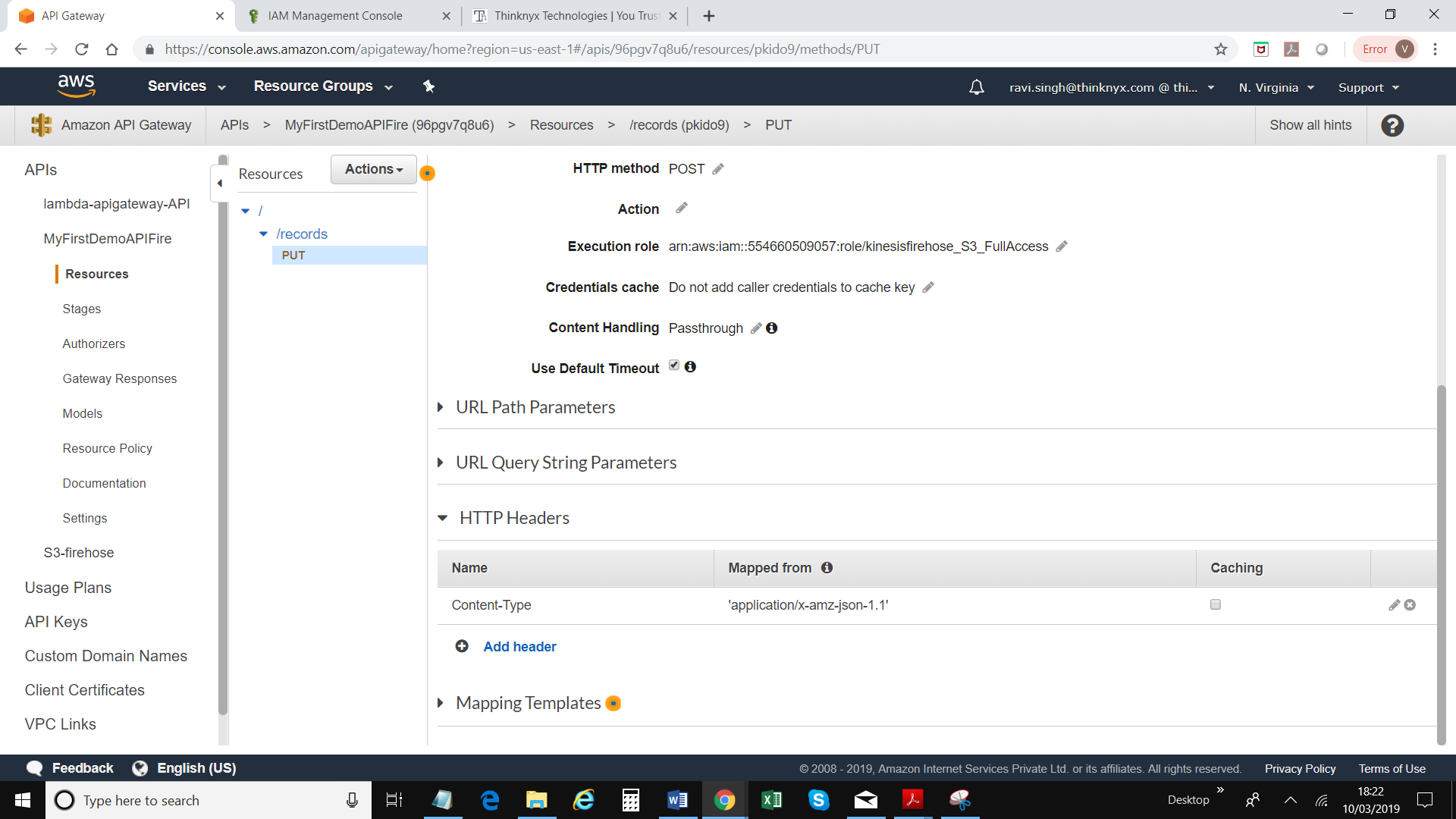


1. Click Save
2. Click on Integration request

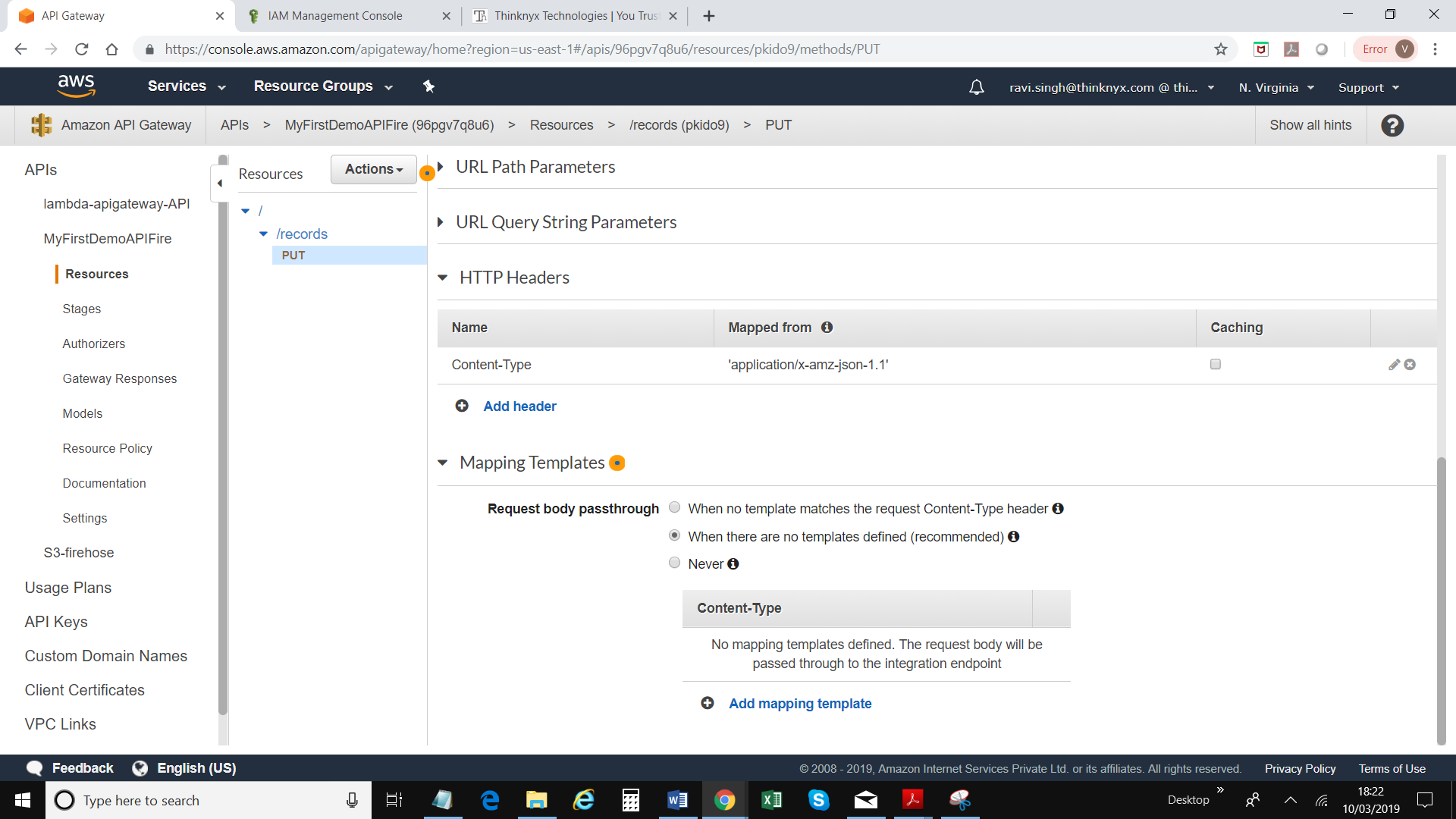


1. Scroll down and click on “HTTP Headers” section and click Add header

Enter Name = Content-Type   
Mapped From = 'application/x-amz-json-1.1'



1. Click on Mapping Template; select “When there are no templates defined (recommended)” under “request body passthrough” field

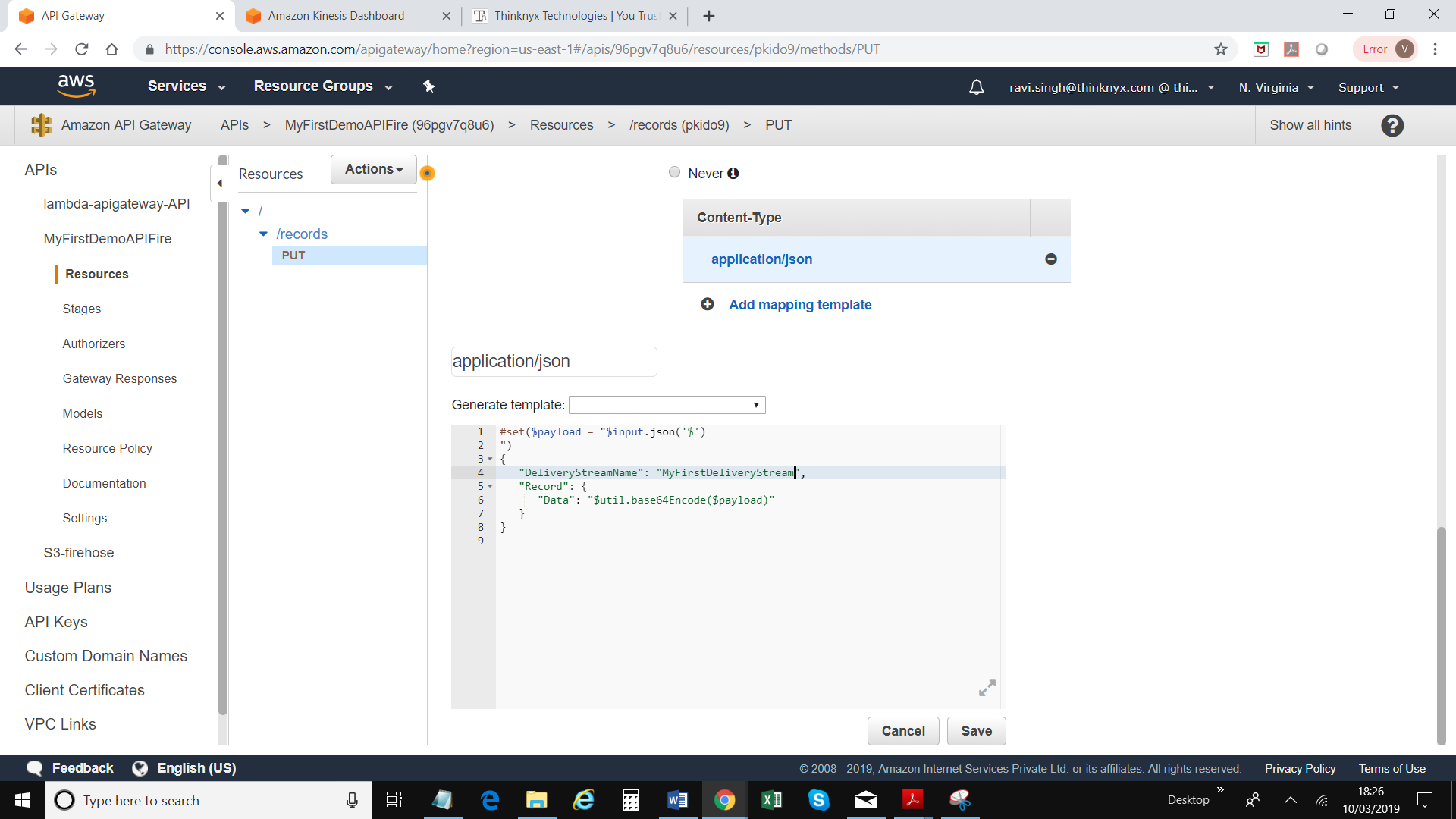


1. Click on Add mapping template
2. Enter “application/json” under Content-Type;

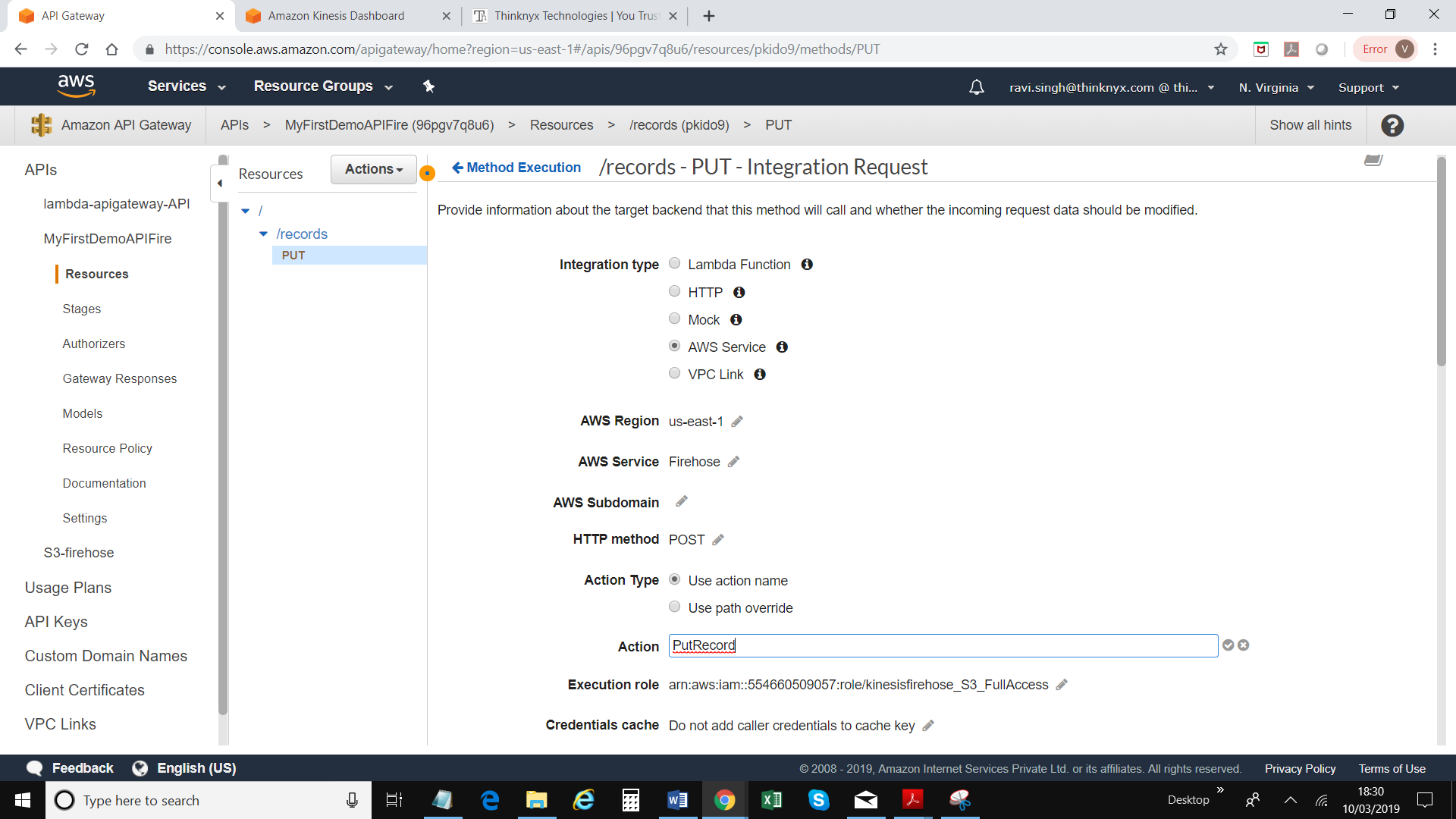
Enter following

“#set($payload = "$input.json('$')  
")  
{  
   "DeliveryStreamName": "MyFirstDeliveryStream",  
   "Record": {   
      "Data": "$util.base64Encode($payload)"  
   }

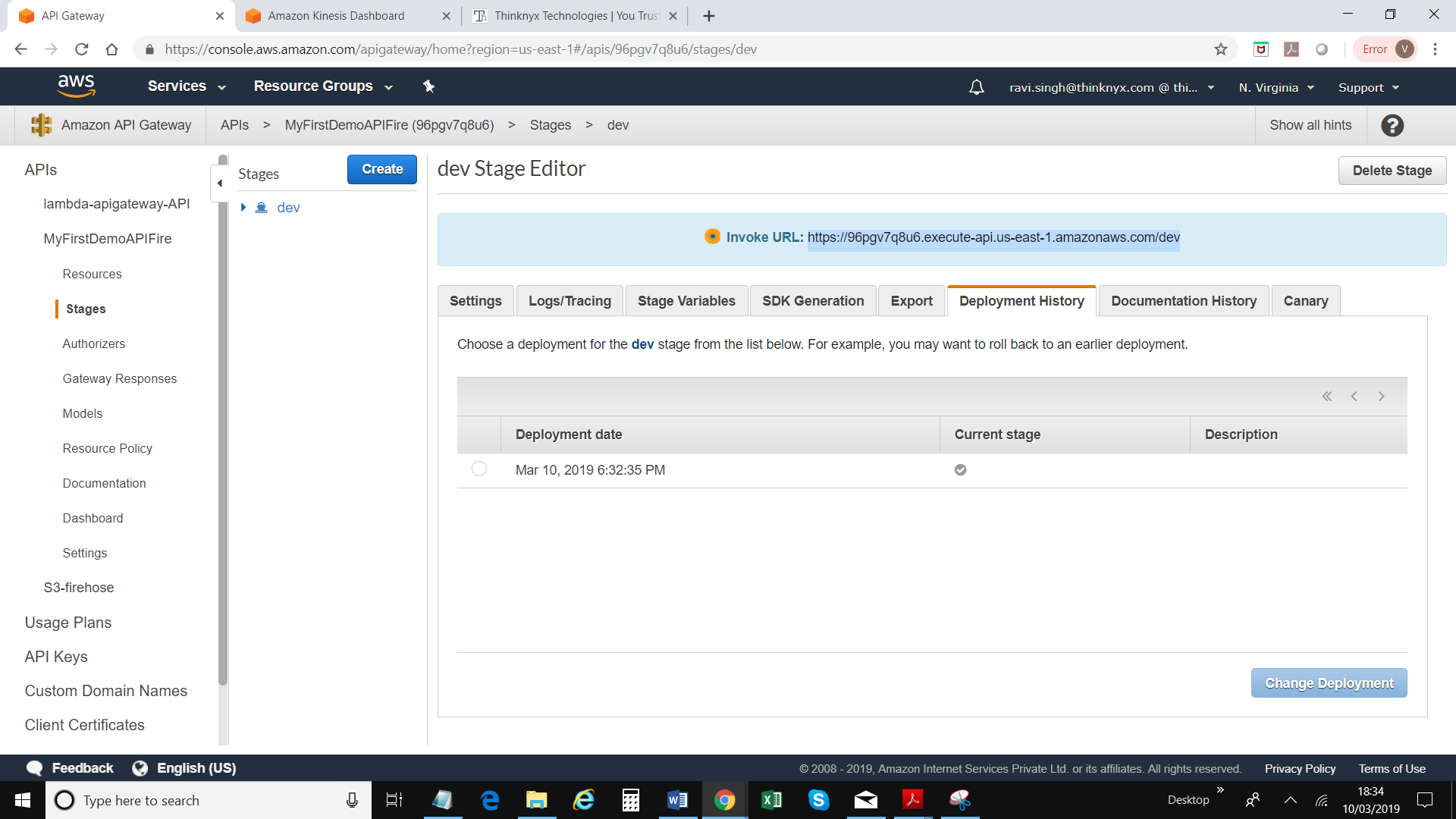
}



1. Click Save
2. Click on Actions => Deploy API



1. Select “New Stage”
2. Type Stage Name, Stage Description & Deployment Description
3. Click Deploy



1. A post record was sent using **Invoke URL:** <https://96pgv7q8u6.execute-api.us-east-1.amazonaws.com/dev>

