**G. Manikanta Reddy**

**Send S3 Event Notification Email Using Lambda And SES**

INTRODUCTION:

Each contemporary application must provide email alerts. In this article, we'll go through how to use AWS Lambda and Amazon SES to send email alerts to users about the addition of new items to an S3 bucket. We'll also show you how to utilise these two services to send a user an email acknowledging their submission.

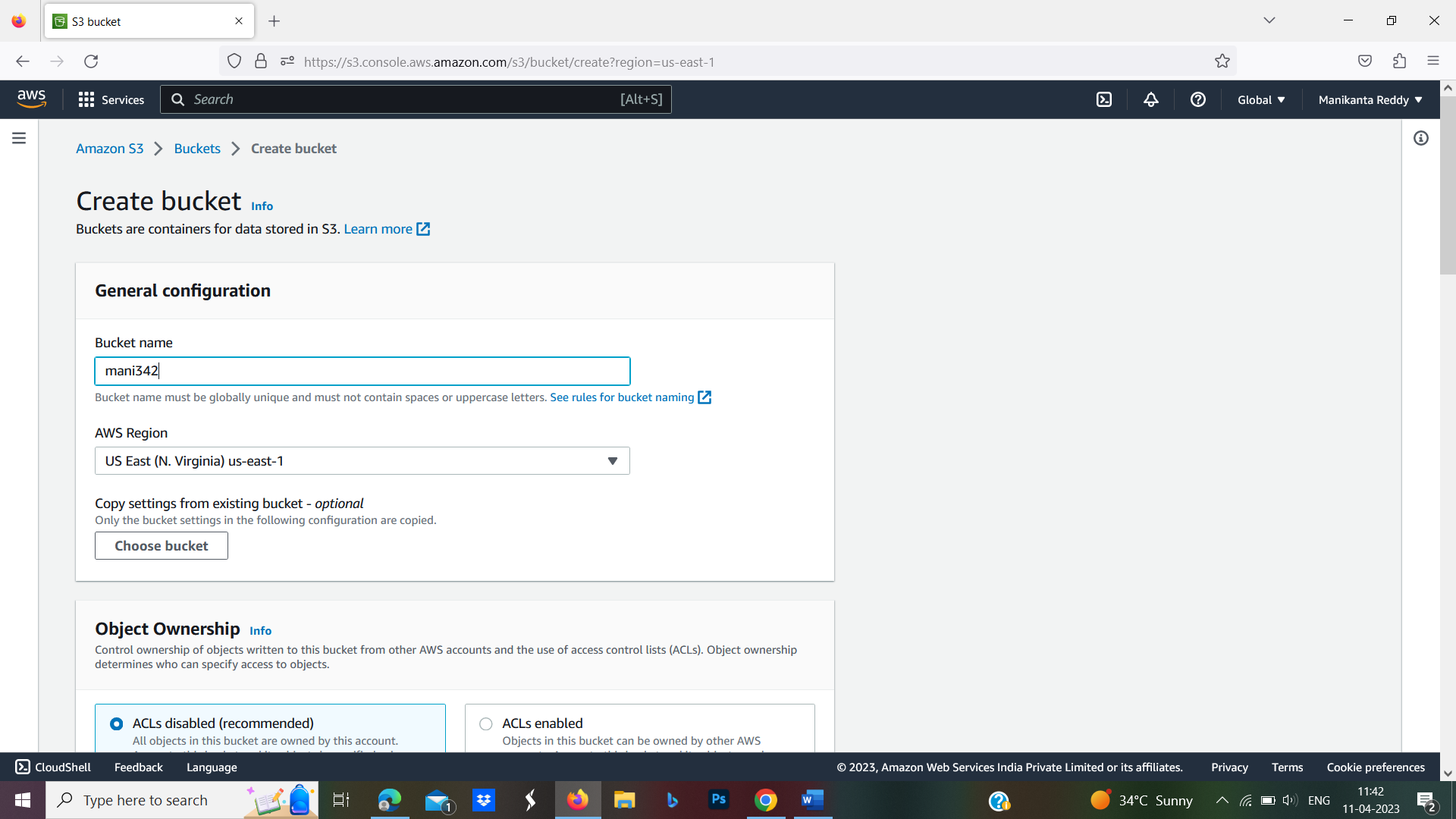
ABSTRACT : -

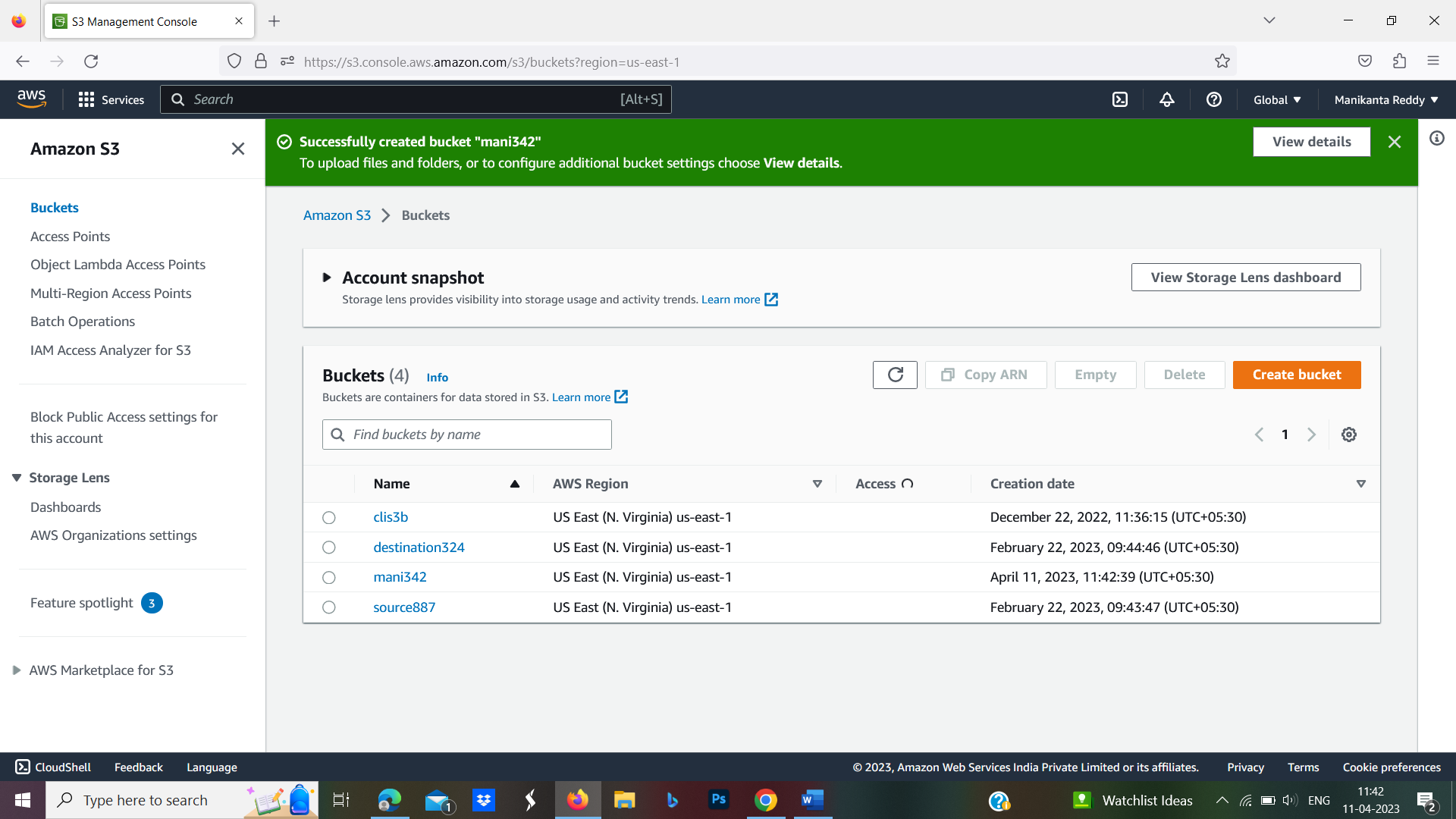
Helping a user to create a notification or an alert when any changes make in amazon services . If a user can import an object or any file into s3 bucket(examplen service) it will send alert by using ses service then user will take precautions about the confidentiality on s3 storage bucket.

To implement this, we will follow the below steps:

1. Create an S3 bucket (or use an existing bucket)
2. Create an IAM policy and an execution role (you can either create it by going to the IAM section on the console or you can choose the option of ‘create a default execution role with basic settings’ while creating a Lambda function)
3. Verify SES identity Domain OR Email – Either a domain bought from an external source or bought from AWS Route53. Email verification using the SES console
4. Create a Lambda function to write code for sending an Email using SES
5. At last, a trigger to the Lambda function with S3 Bucket as the source initiates its execution while adding a new object.
6. Add a new object in the S3 bucket and test the entire implementation to check how
7. Go to the S3 Console and click on **Create Bucket** as shown in the below

Step-1:Enter a name for the bucket. You can disable the ‘Block all public access’ checkbox if you want your bucket to be public. Also, you can add tags for objects added in the bucket or you can enable Bucket Versioning if required.

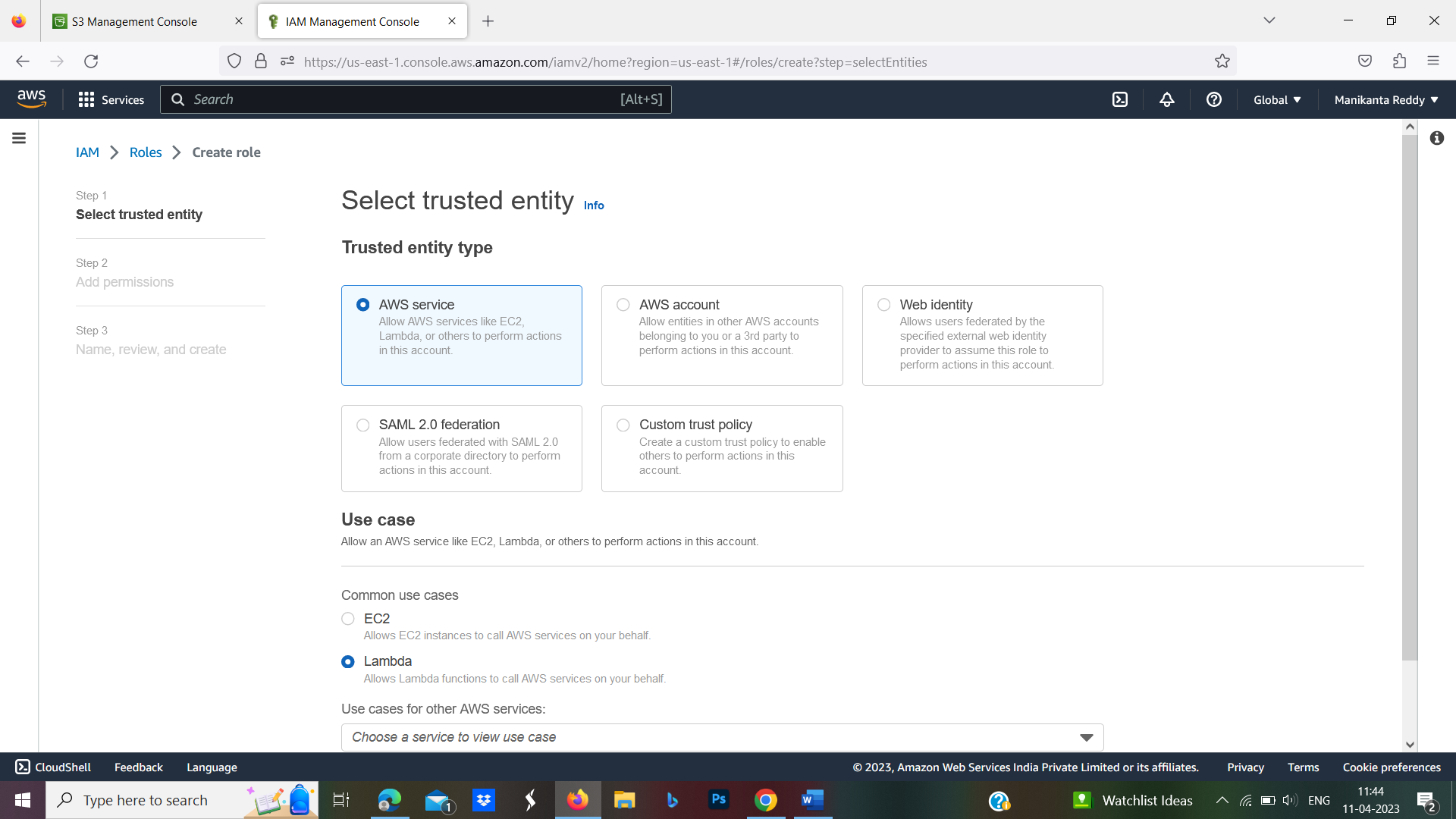


Step-2: We are not enabling Bucket versioning and not adding any tags as of now. Click on **Create Bucket** and we will see the bucket created.

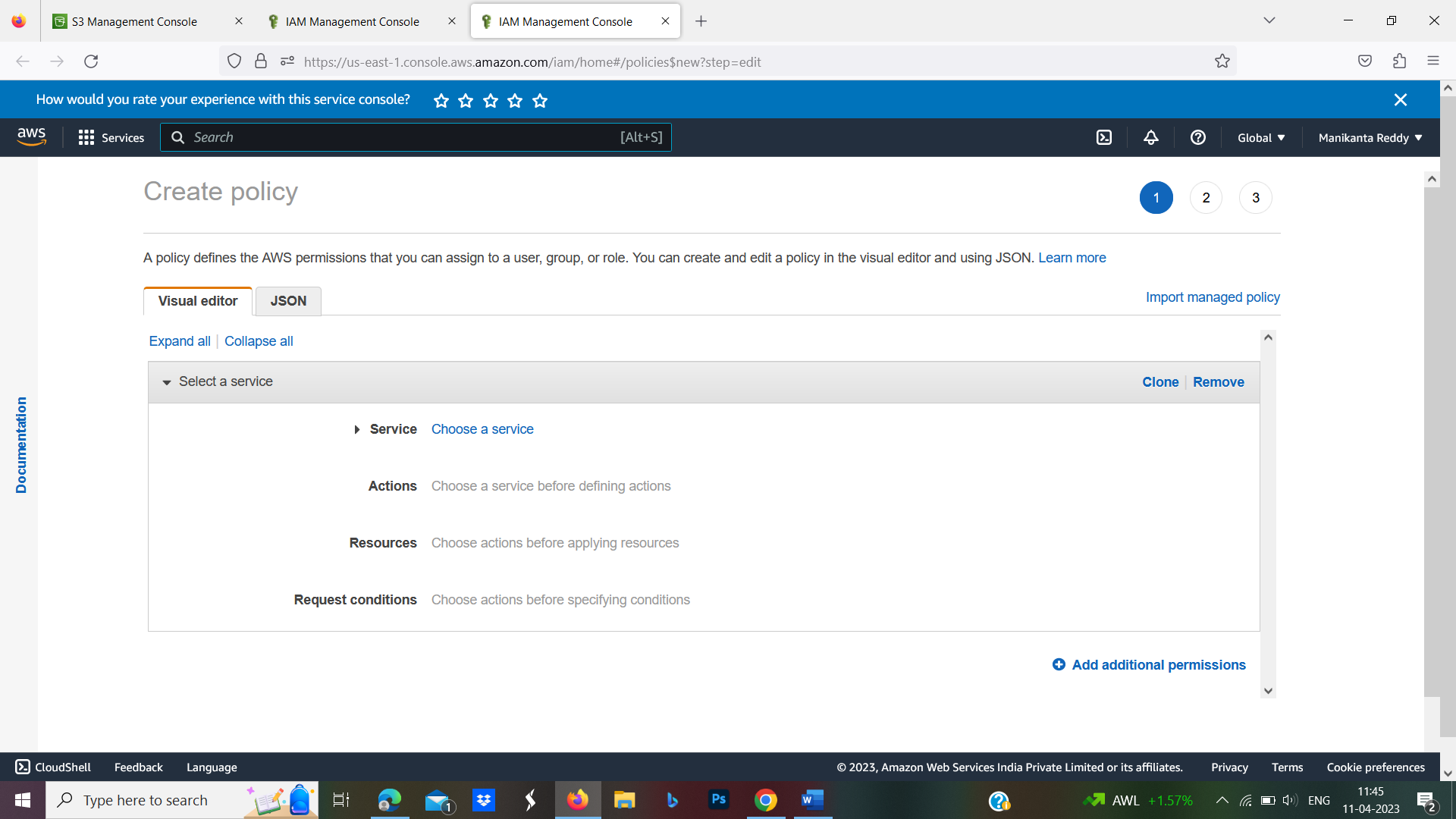
Step-3:Now, go to the IAM Console on AWS.



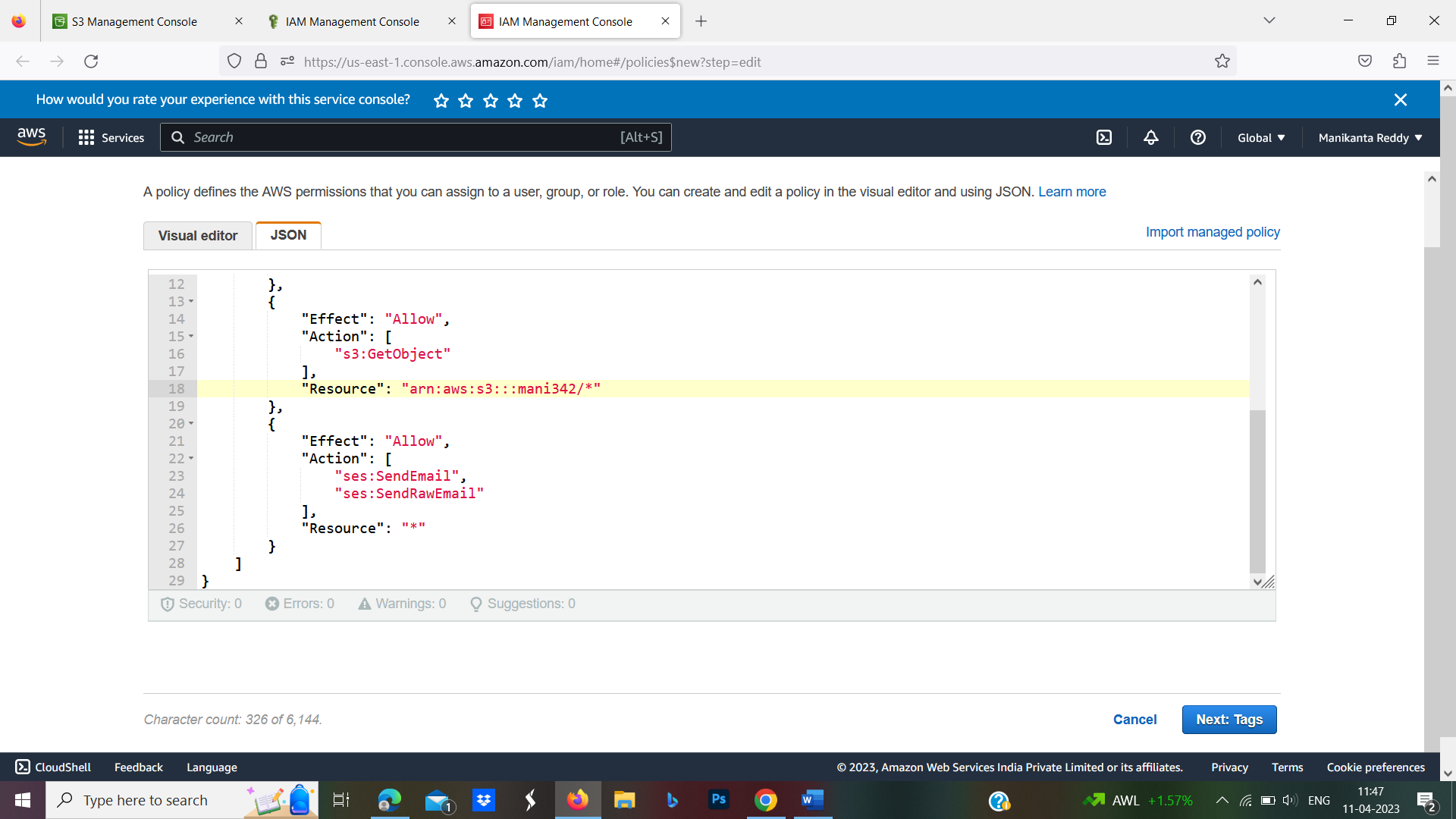
Step-4:Under roles, create on **‘Create role’**, choose **‘Lambda’** and click on **Next:Permissions.**



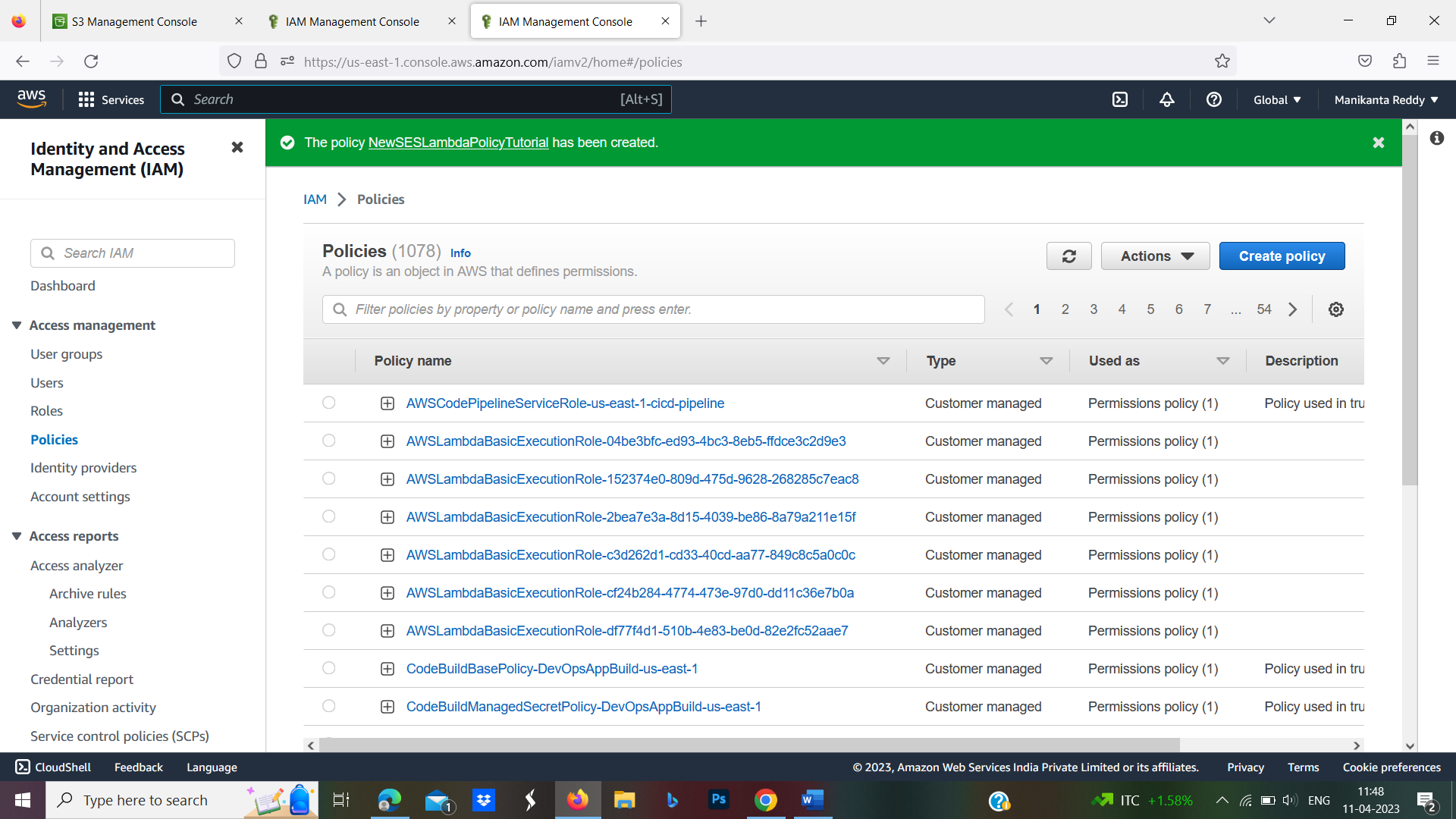
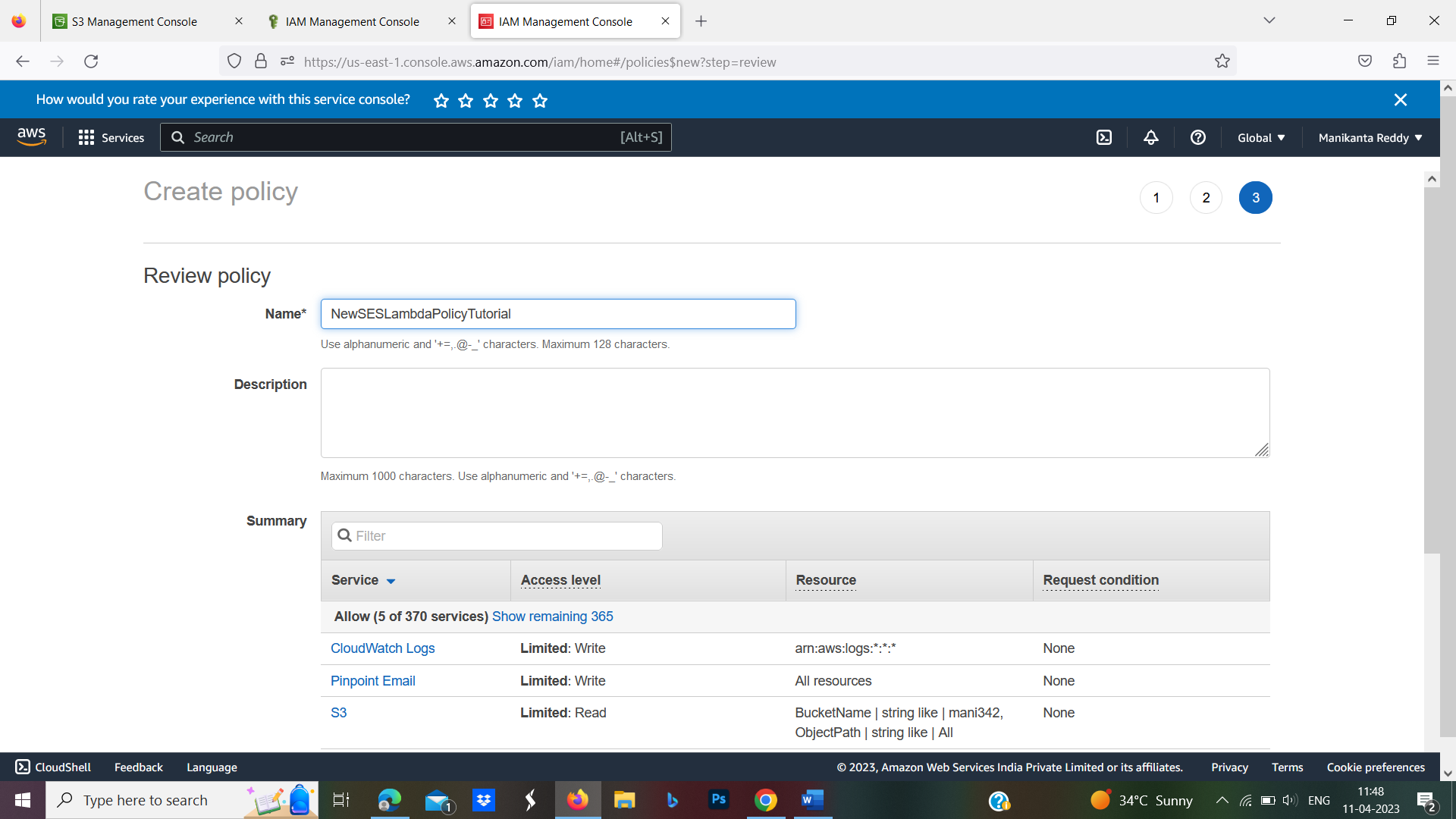
Step-5:Based on the AWS best practices, it is suggested that you create a policy with only the required permission. Click on **Create policy.**



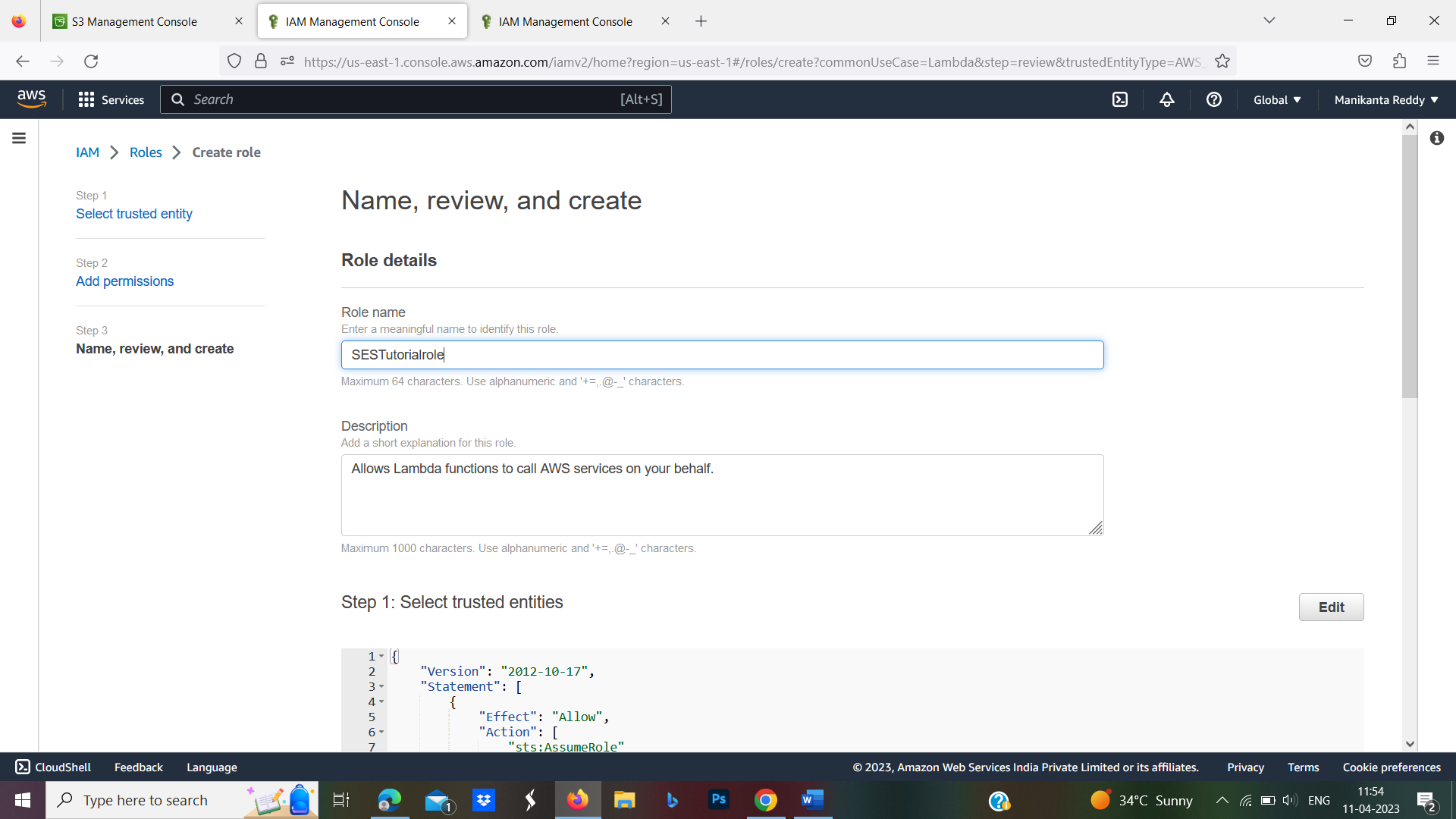
Step-6:Under JSON, add the following JSON as given below. Rename ‘mybucket’ to the bucket name you created above and click on ‘Review Policy’.



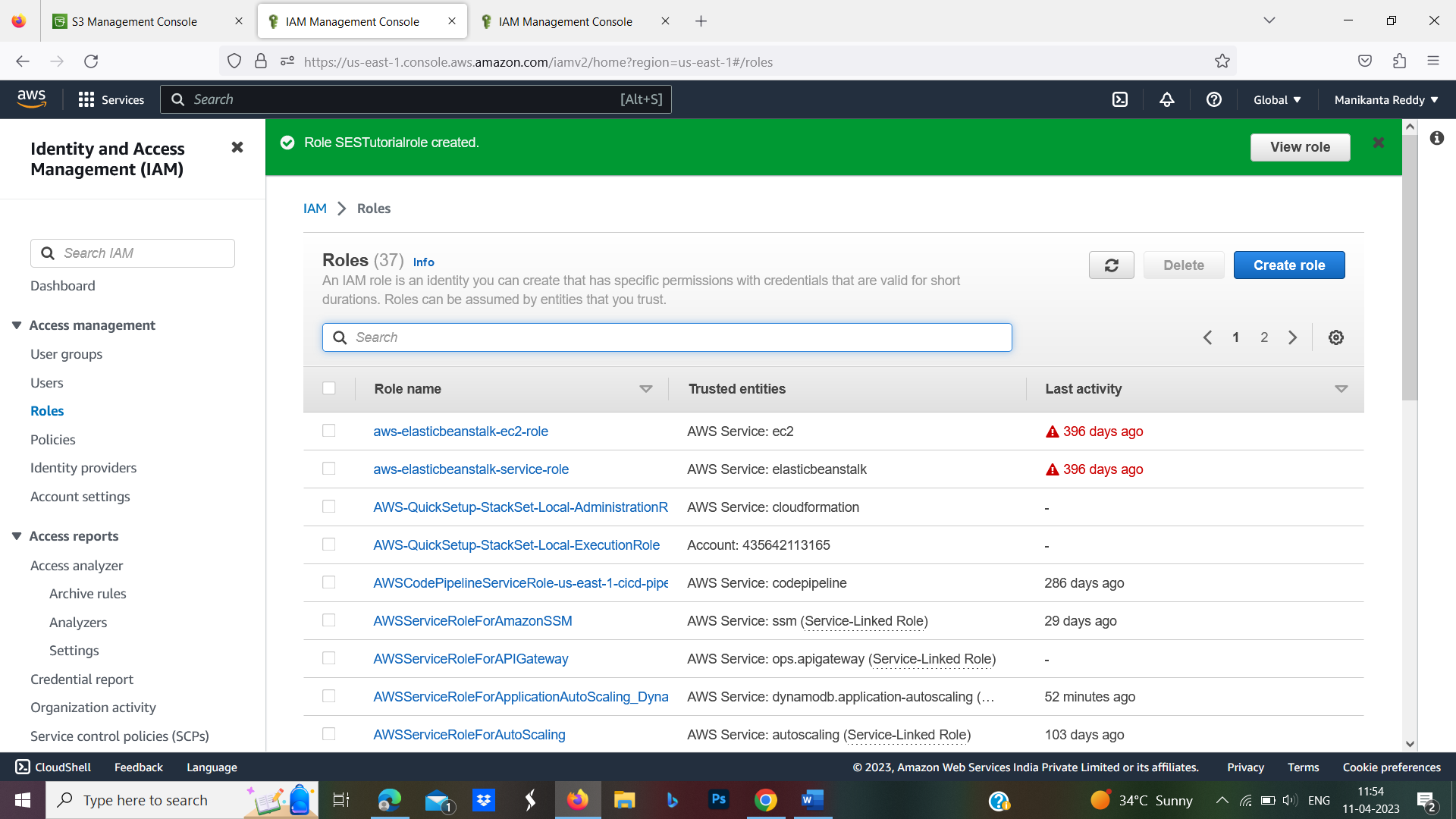
Step-7:Give a name to the policy and click on **Create Policy.**



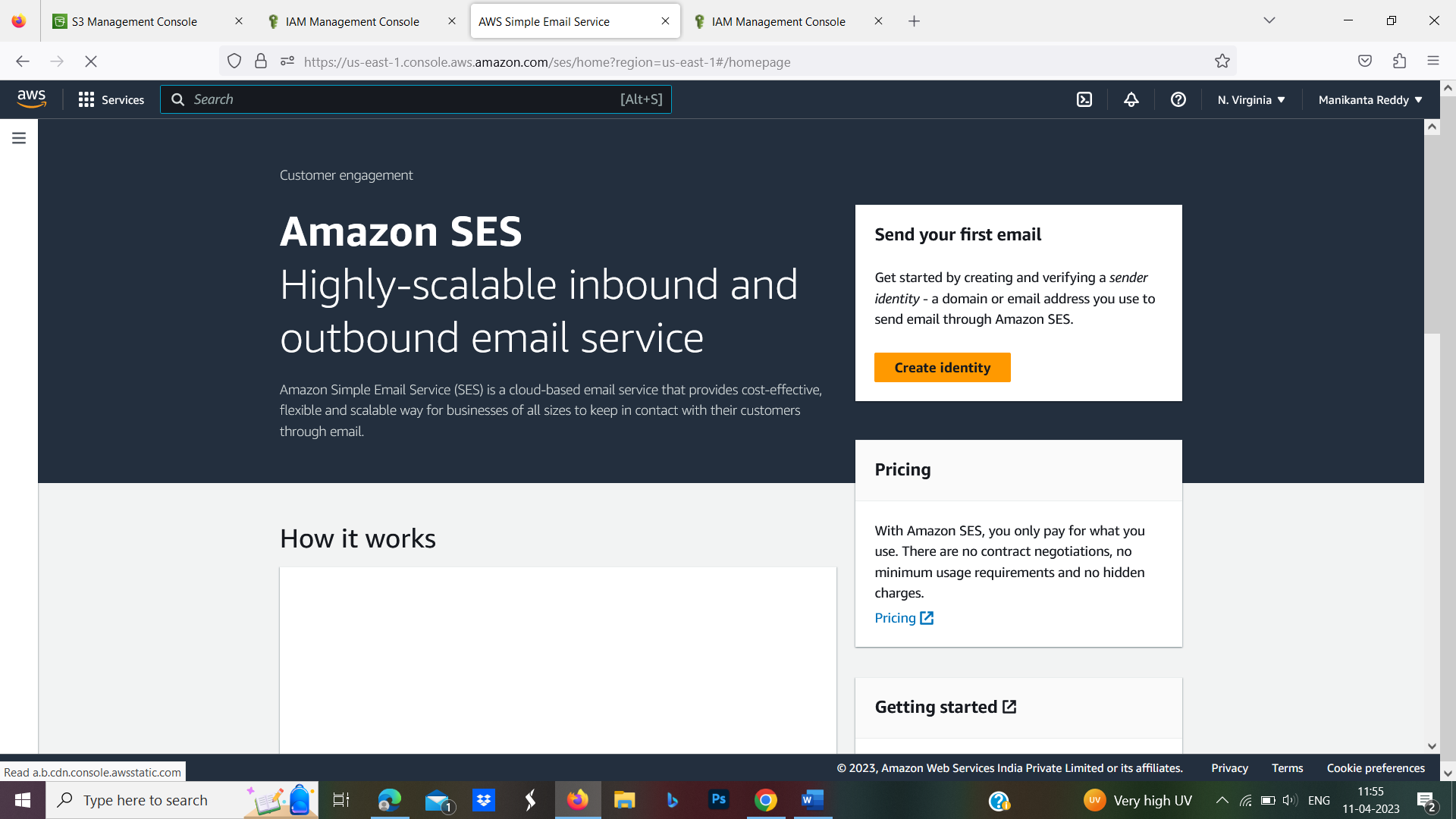
Step-8:Review the policy attached and enter a role name. Once done, click on **create role.**



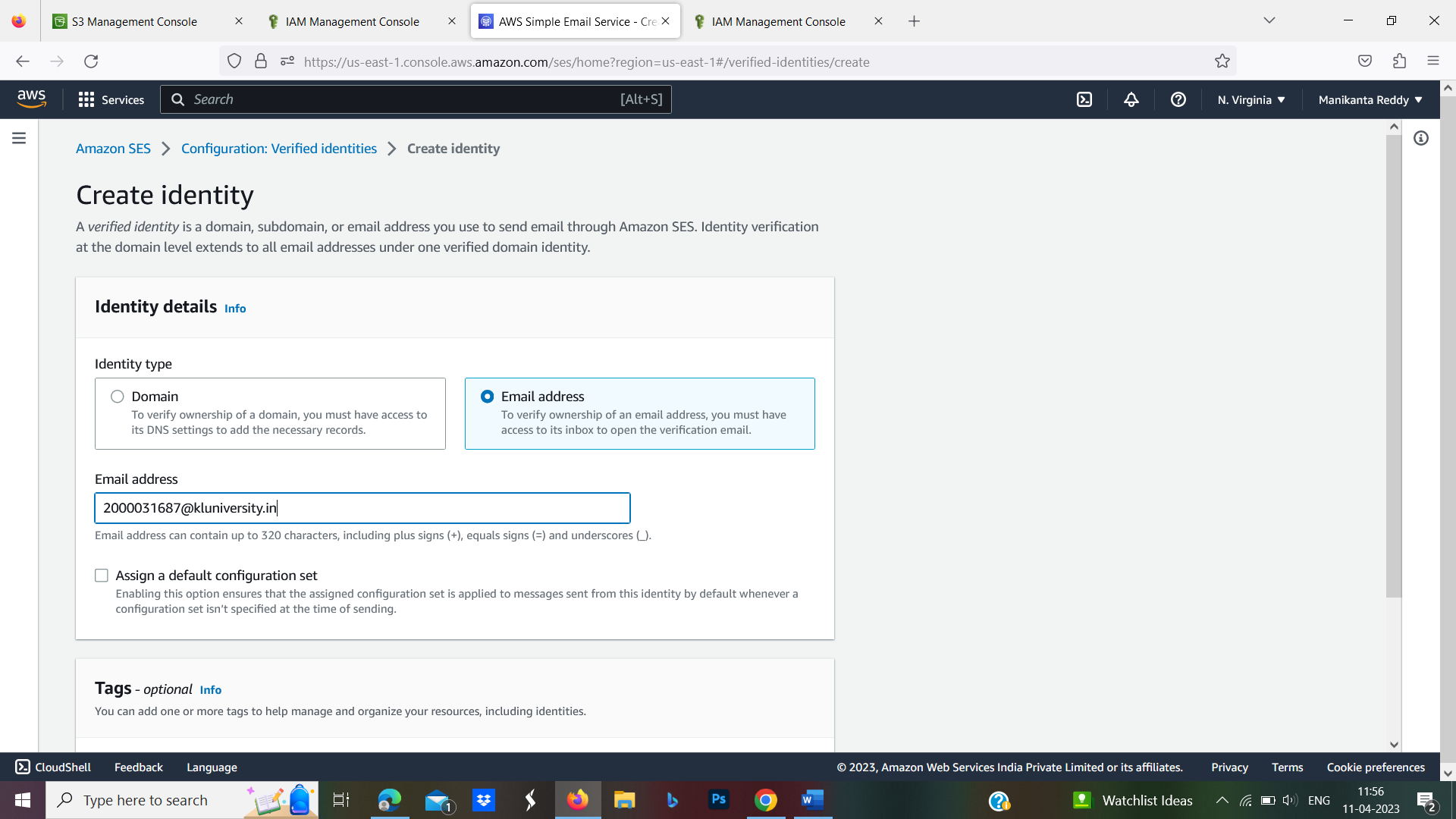
Step-9:The role Which we are created just now it will be visible under the roles section on the console.



Step-10:Now, open the Amazon SES Console .

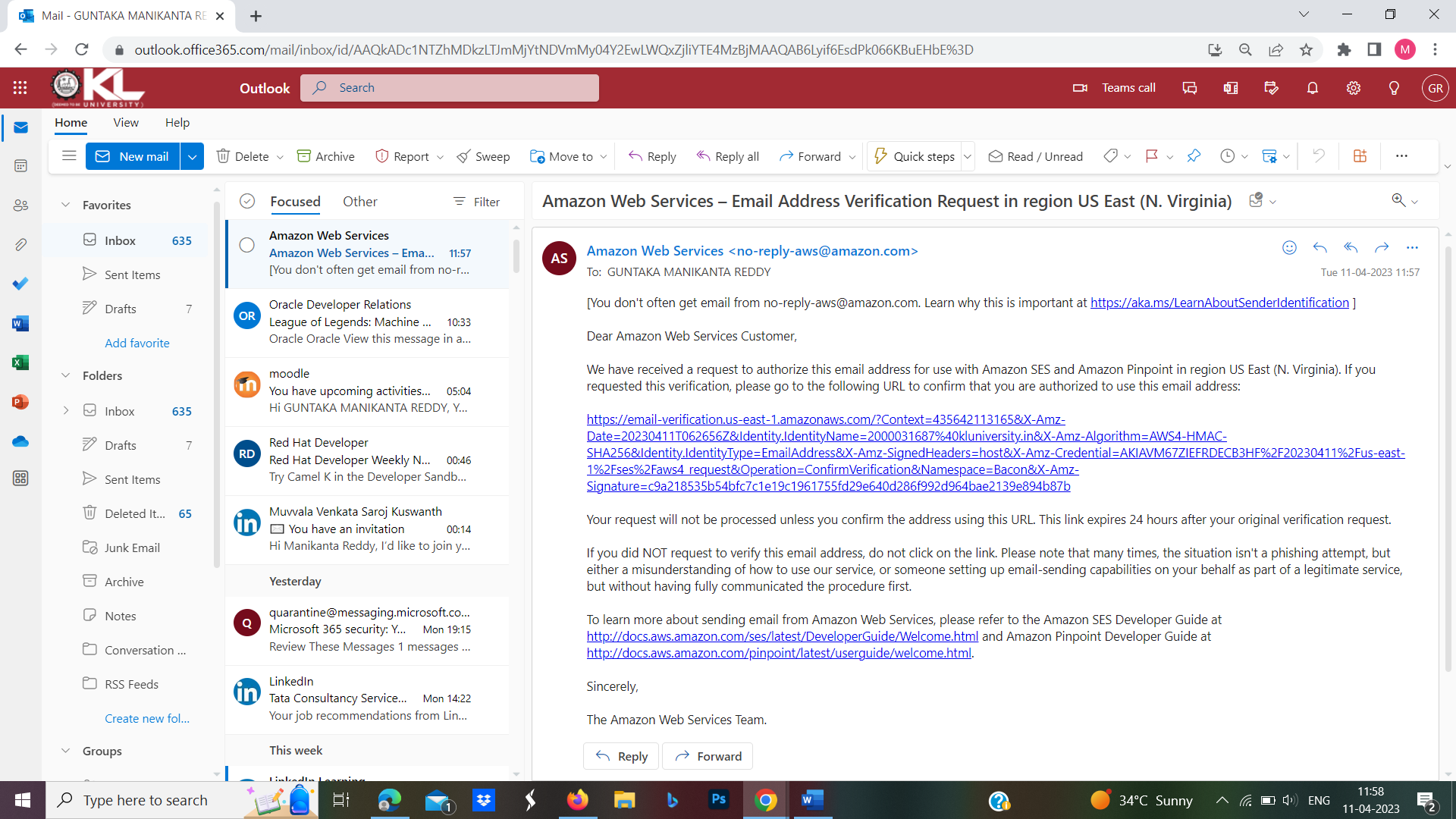
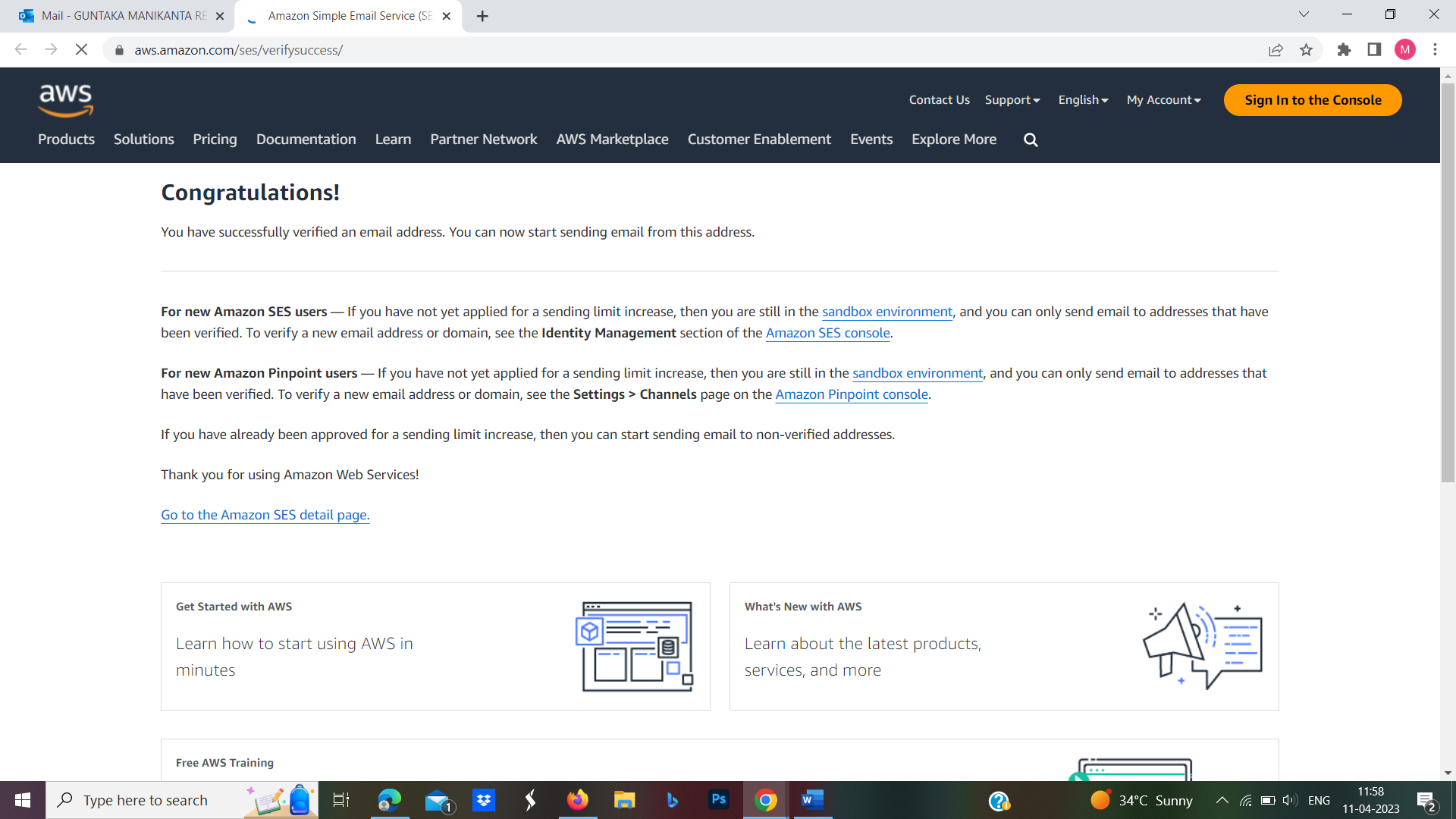


Step-11:Click on the ‘Email Addresses’ under ‘Identity Management’ on the left side.

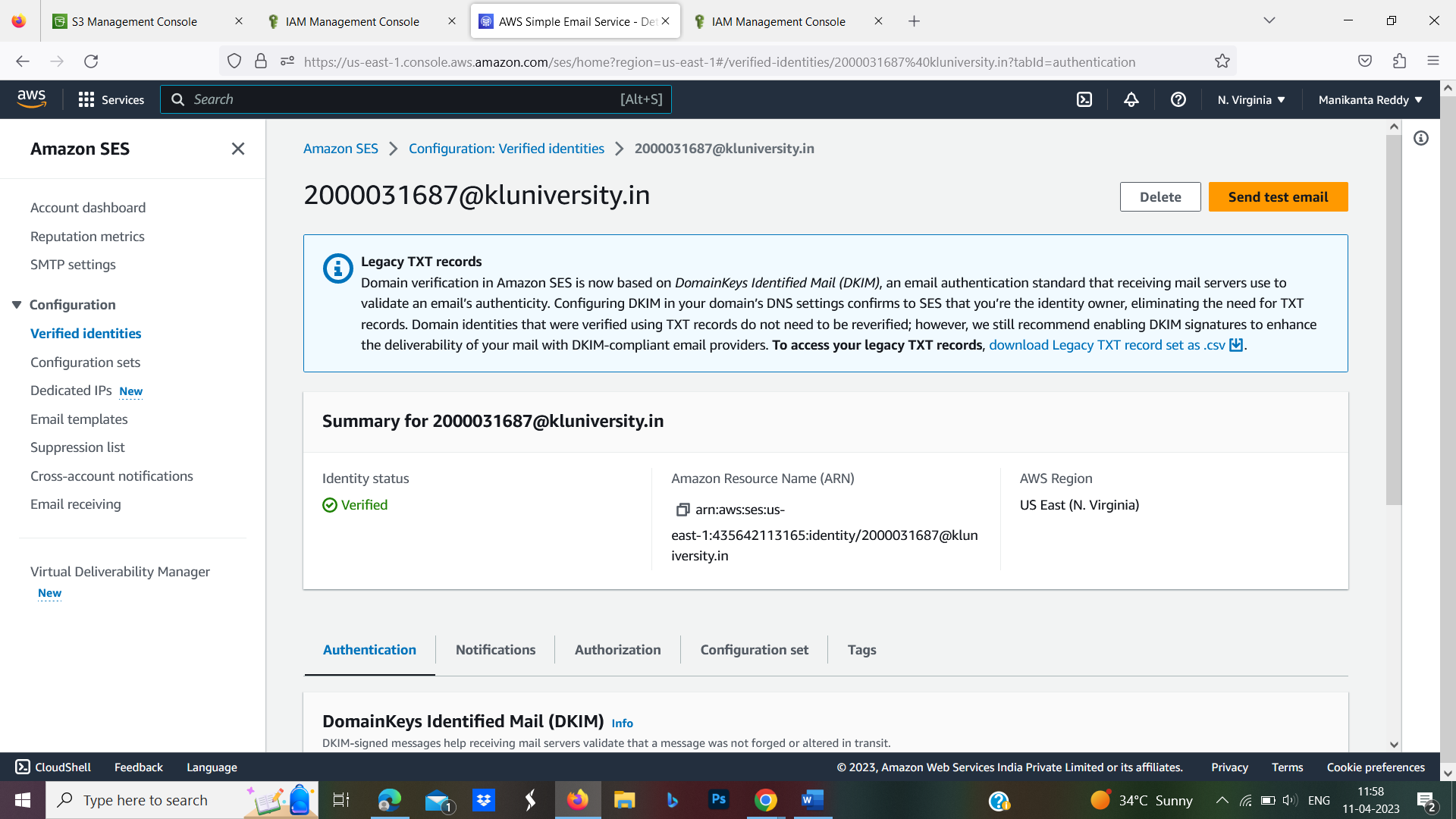


Step-12:Click on **Verify a New Email Address** and a modal will appear. Enter the email on which you want to receive emails. Remember that both the ‘sender’s and receivers’ email addresses will have to be verified to send and receive emails.

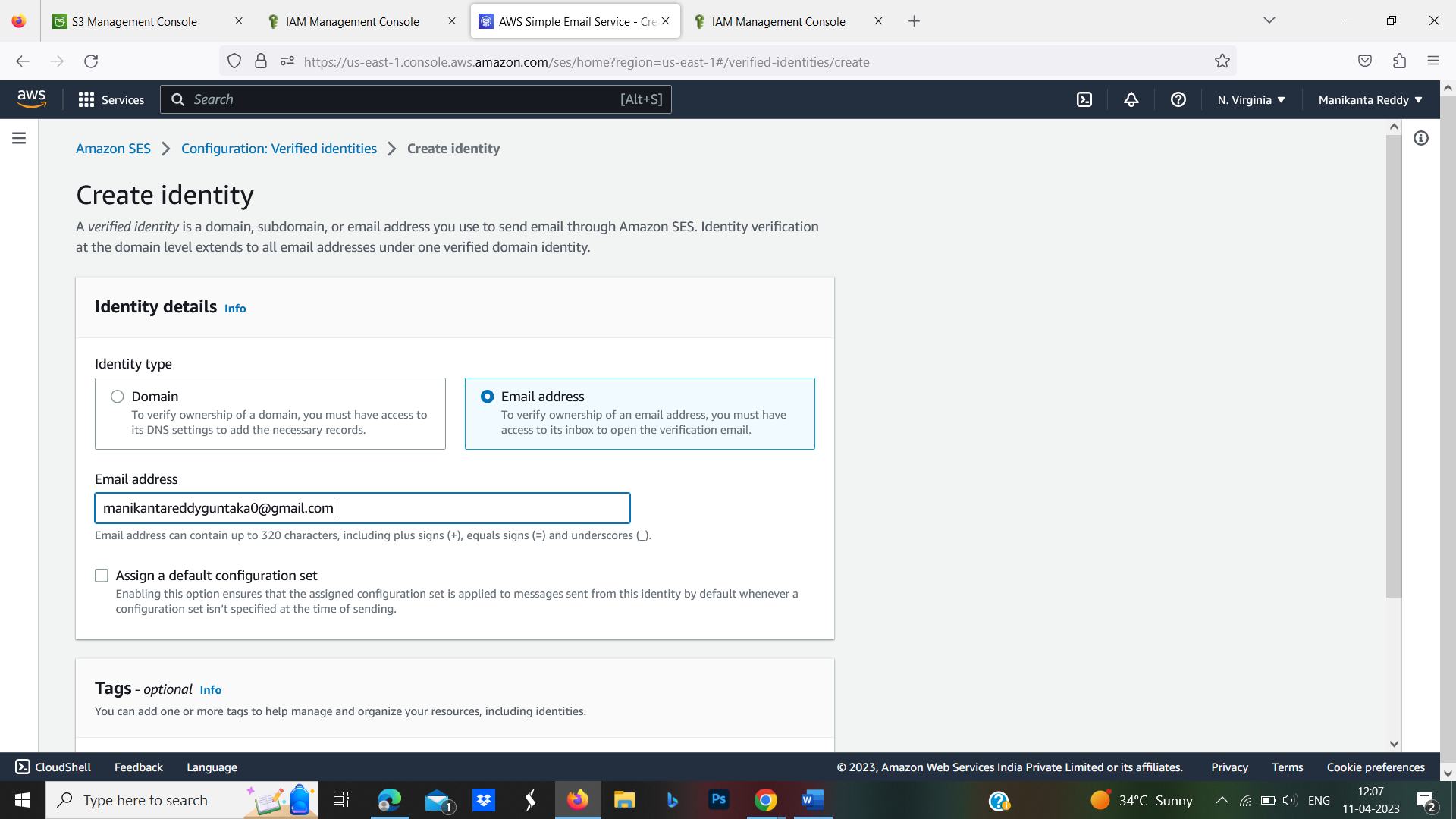
* For this exercise, we will verify one email address and use the same email address as sender and receiver.

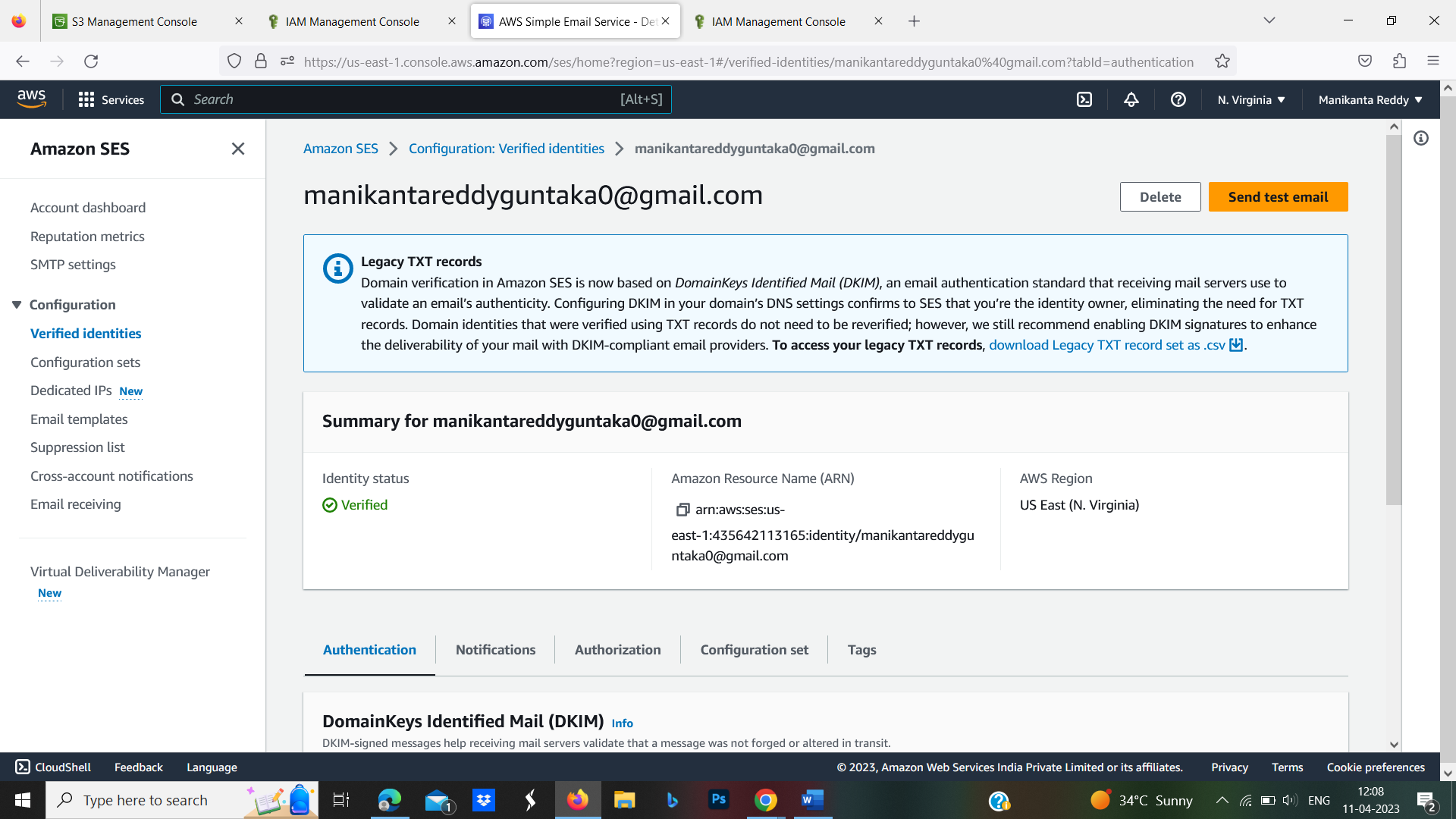
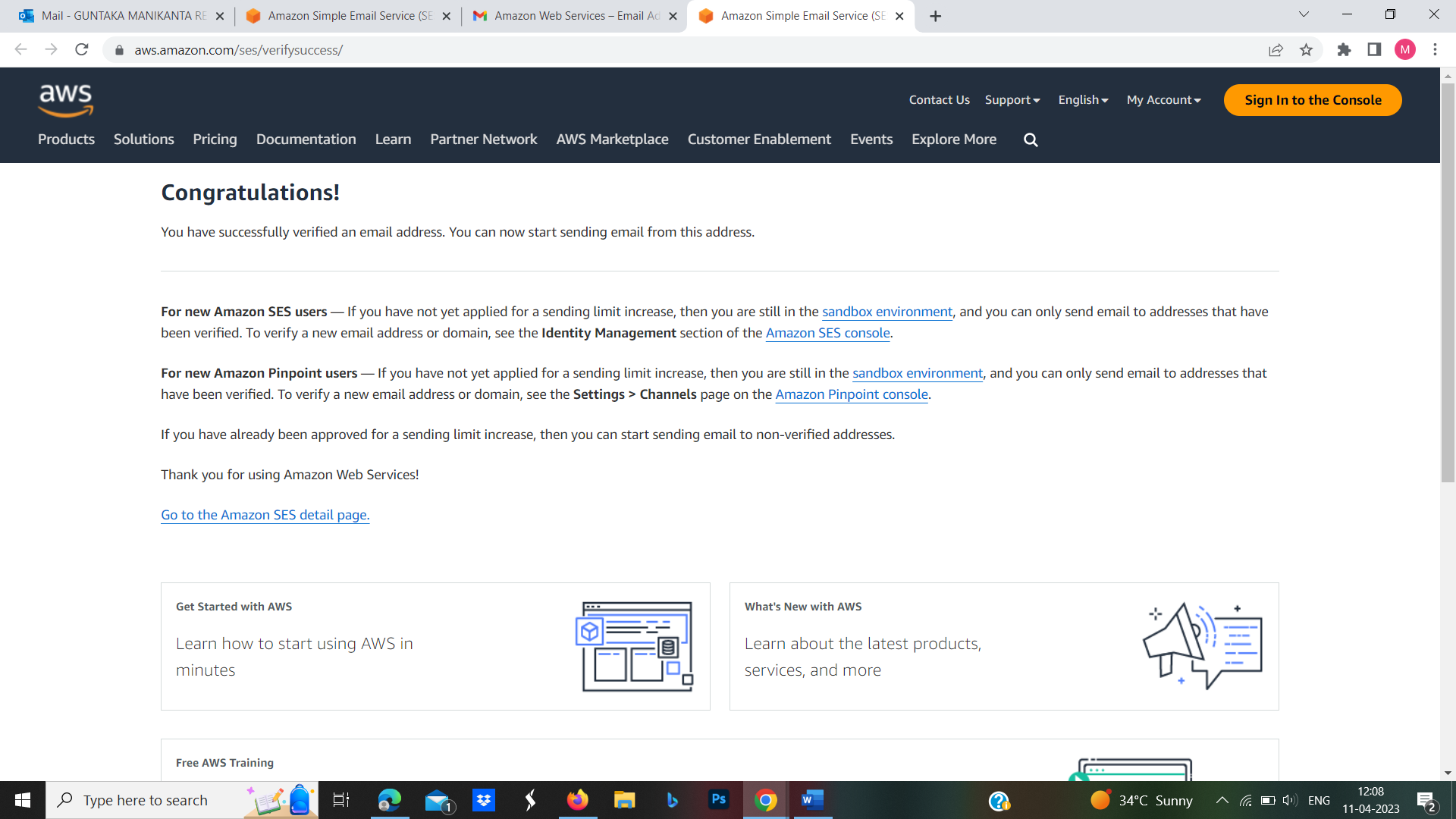
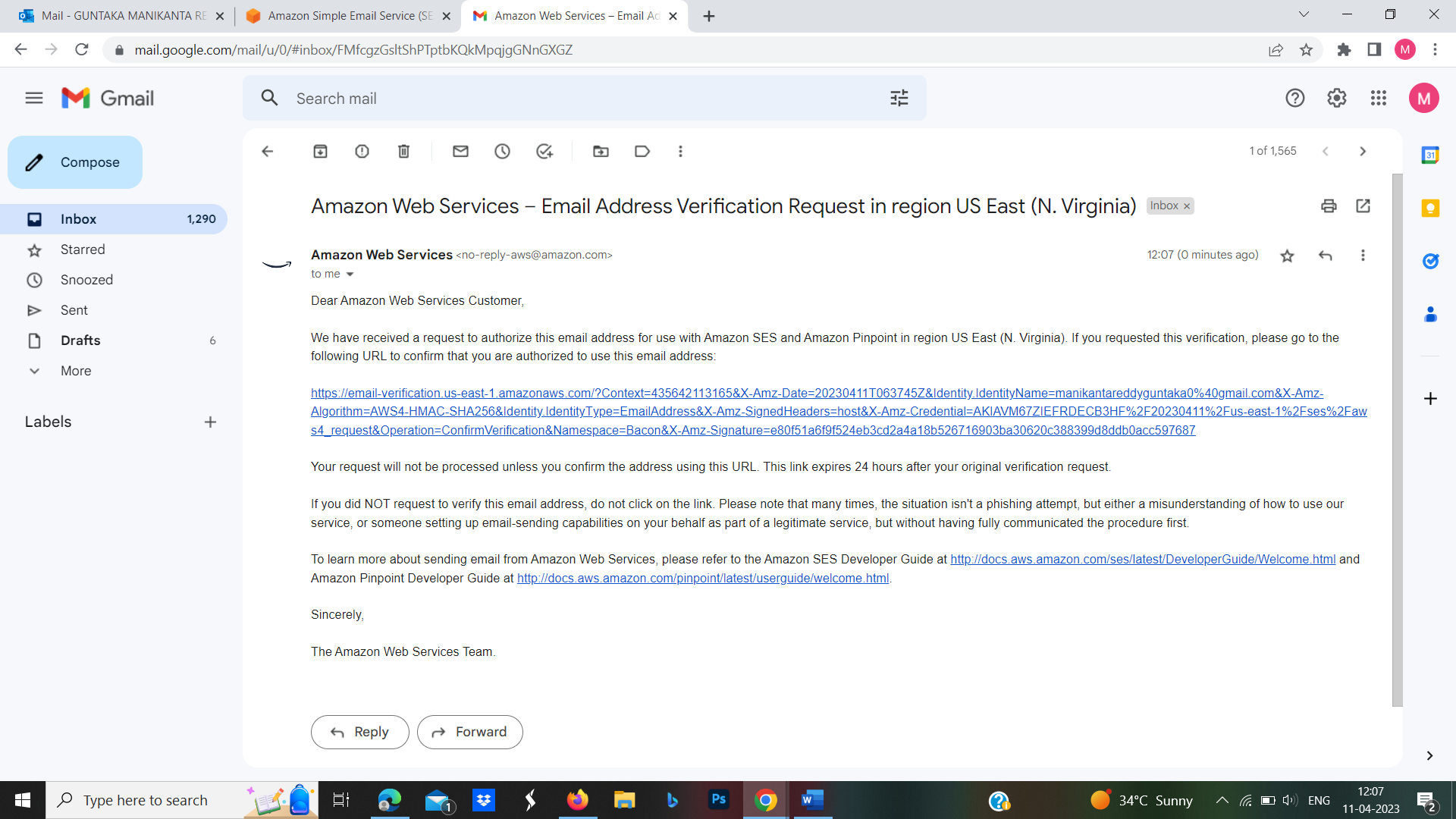


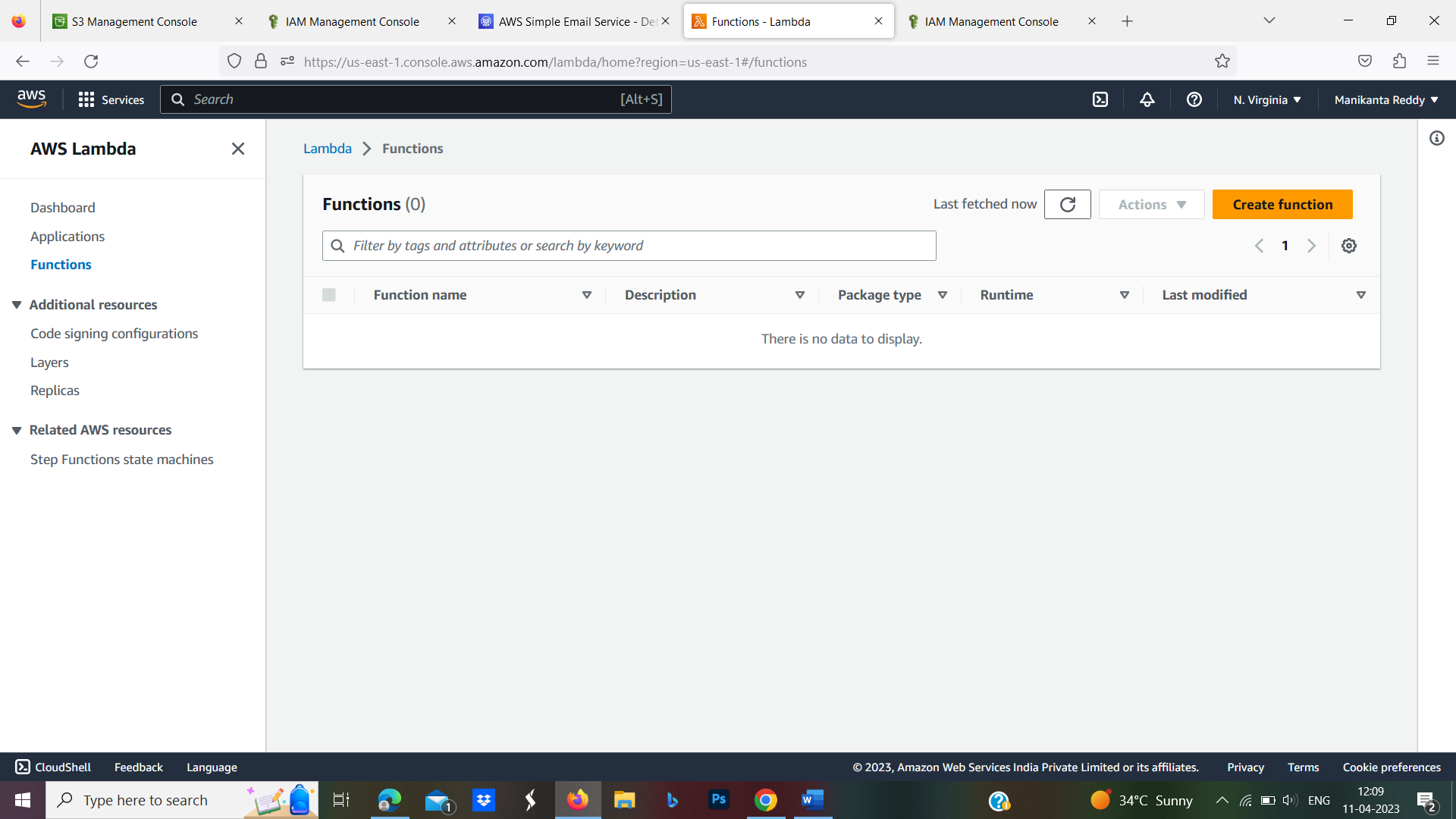
Step-13:On successful, your email address will be verified and will appear on the dashboard with the status verified



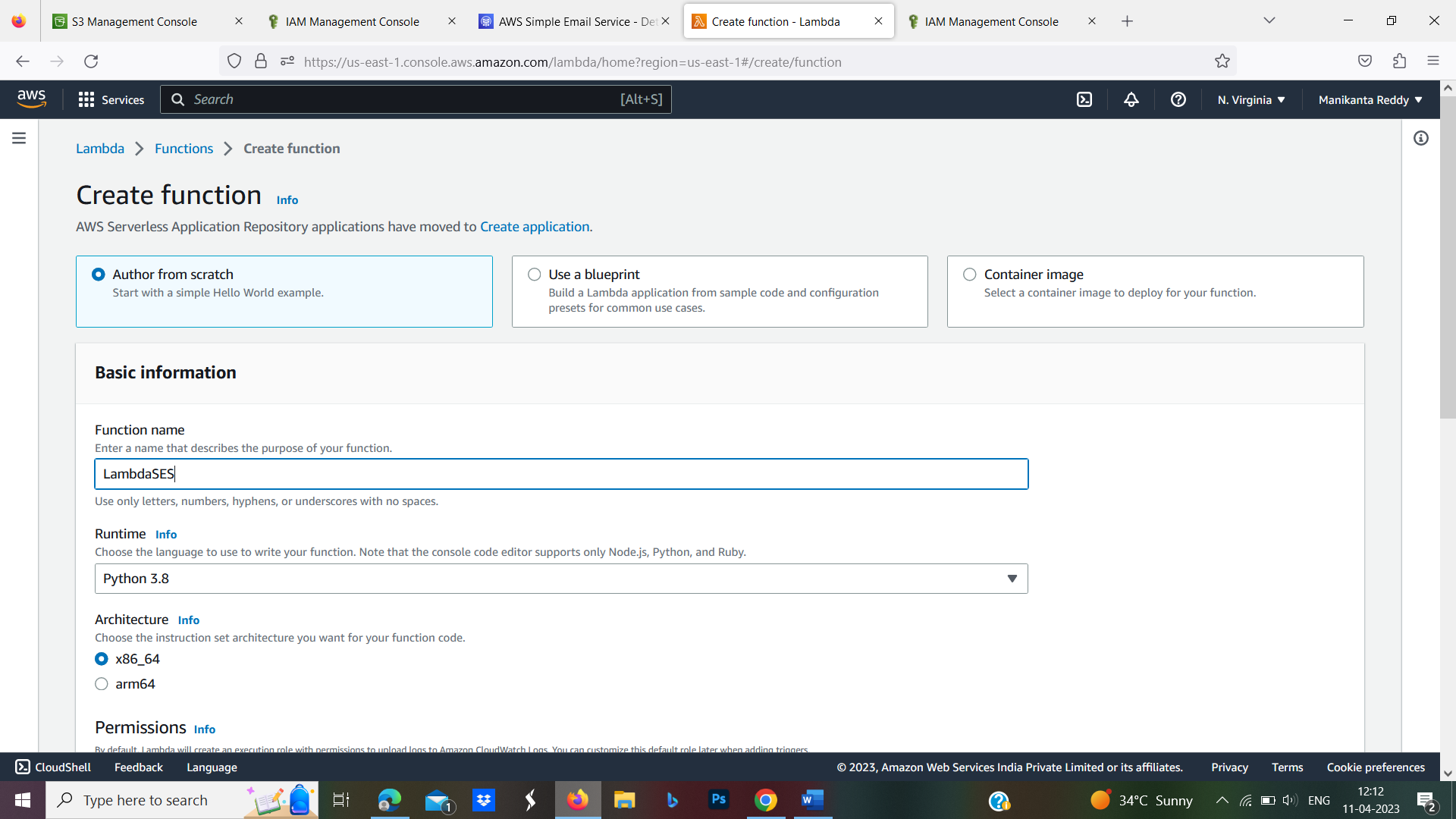
Step-14:Repeact the above process which we are done in step12 and 13 and create another email and verify it.



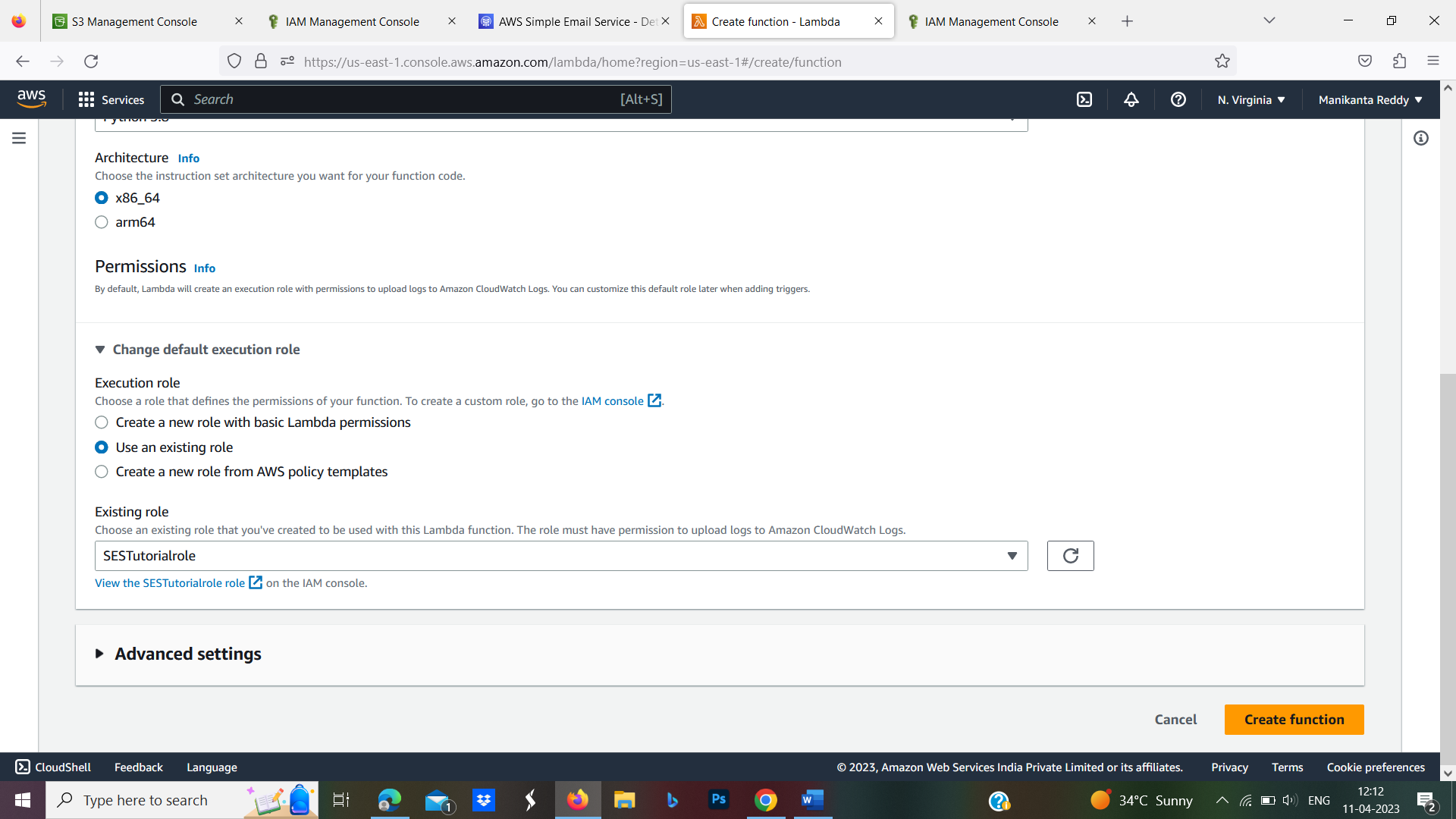


Step-15:Now, open the AWS Lambda console. Select**Functions** mentioned in the left navigation pane .

Step-16:Select **Author from scratch**, enter the function name, select the runtime as **Python 3.8**



Step-17:Under the Execution role, select **Use an existing role** and select the role you created above and click on **Create Function.**

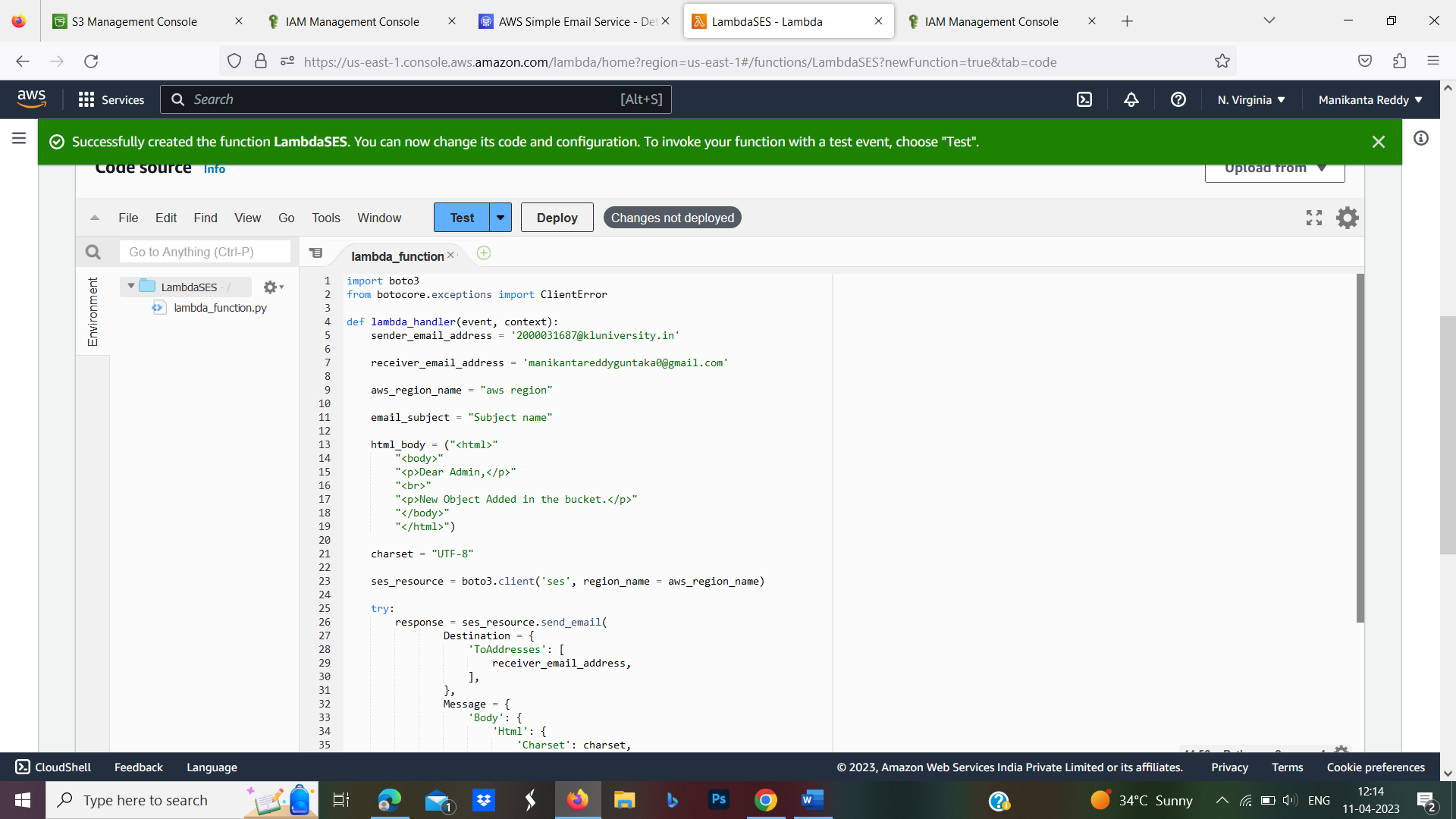


Step-18:Upon clicking on**Create function**, you will be navigated to the function console.



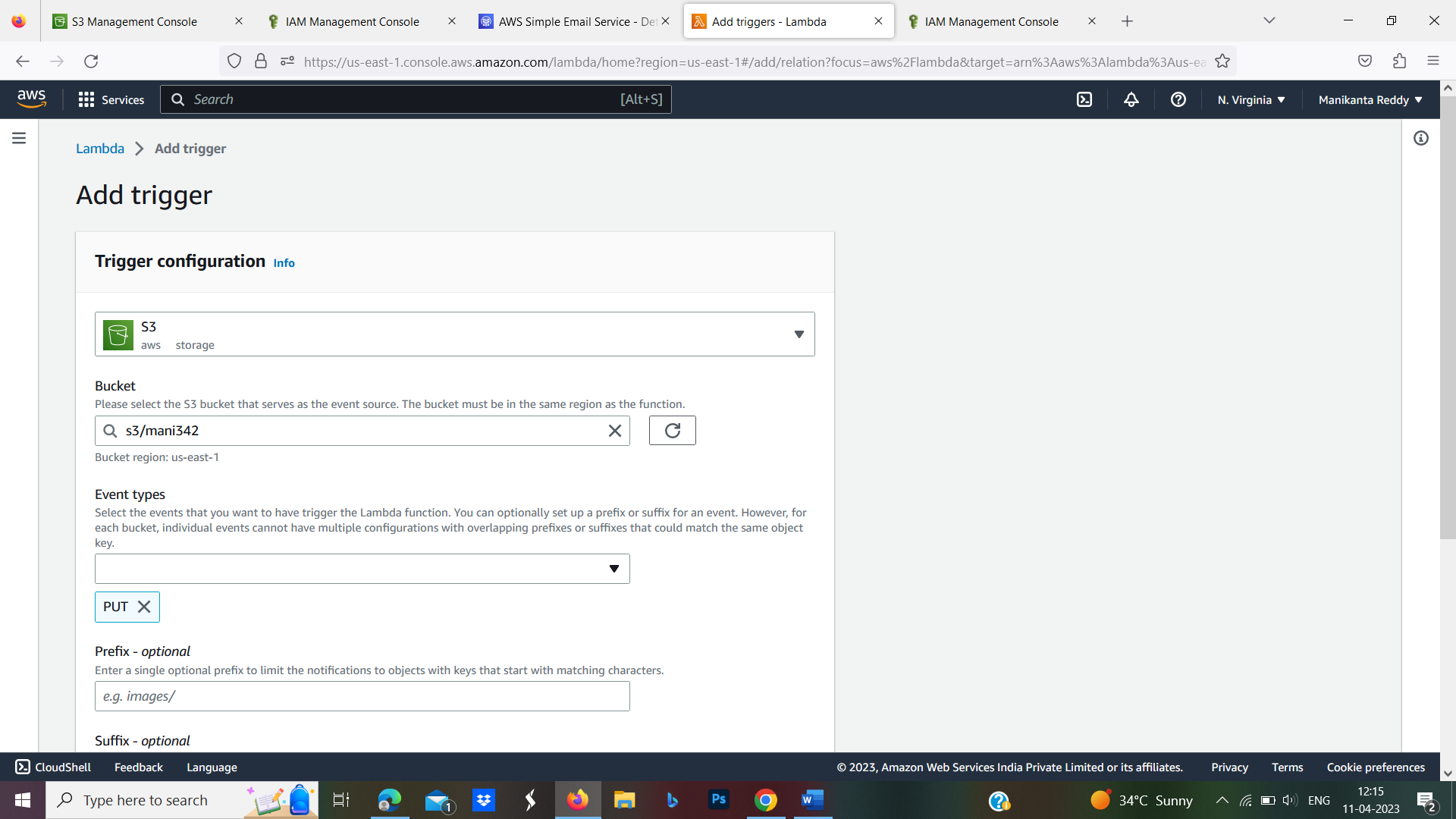
Step-19:Now scroll down and under the function code in the editor pane.

* Change the ‘sender’s email address’ to the email address using which you want to send the email.
* Change the ‘receiver’s email address’ to the email address on which you want to receive the email.
* Ensure that both the email addresses (sender and receiver) are verified on Amazon SES.

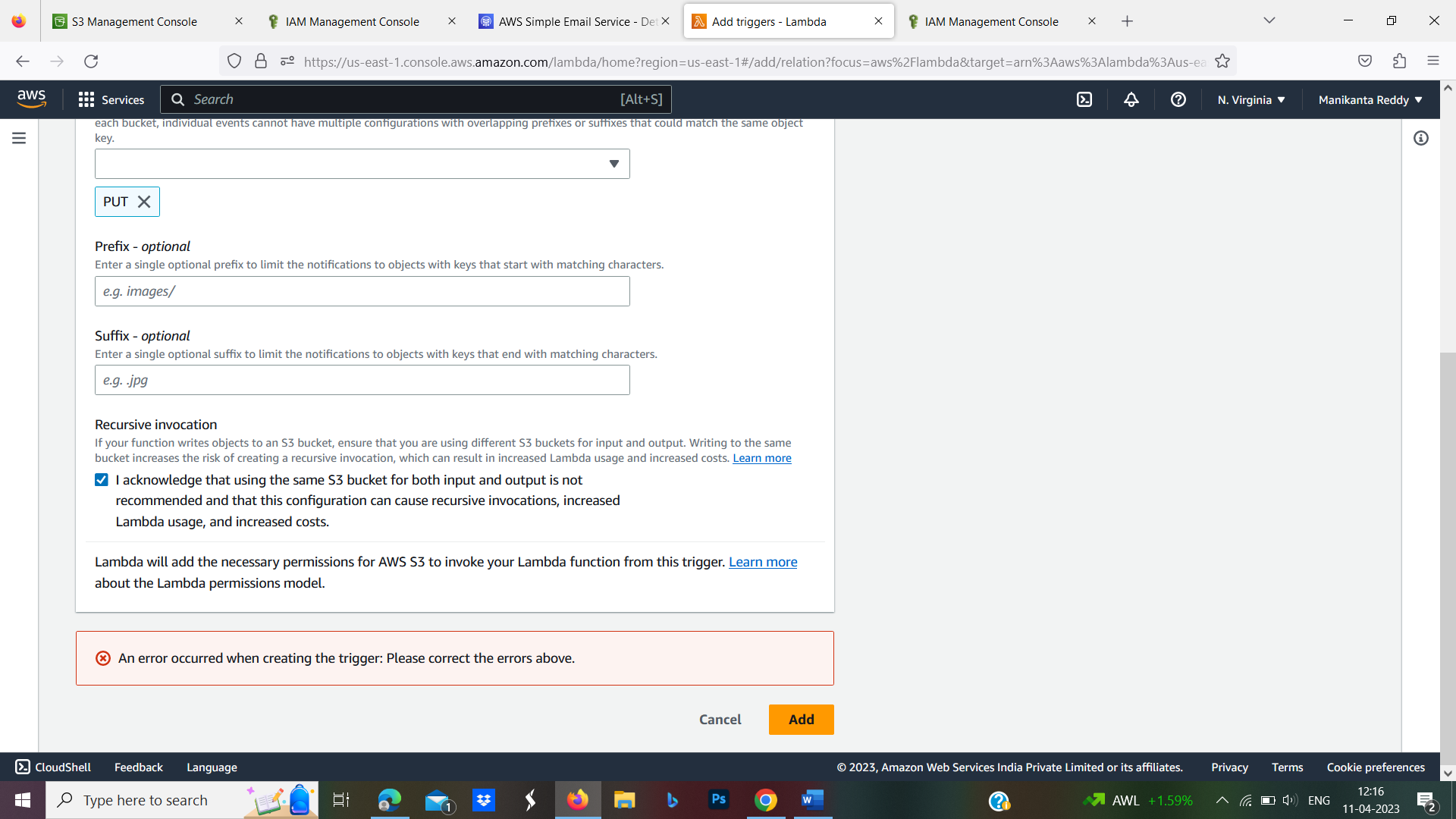


* Change the ‘aws region’ based on the region you are operating in.
* You can edit the ‘Subject Name’ and the ‘HTML Body’ as you want

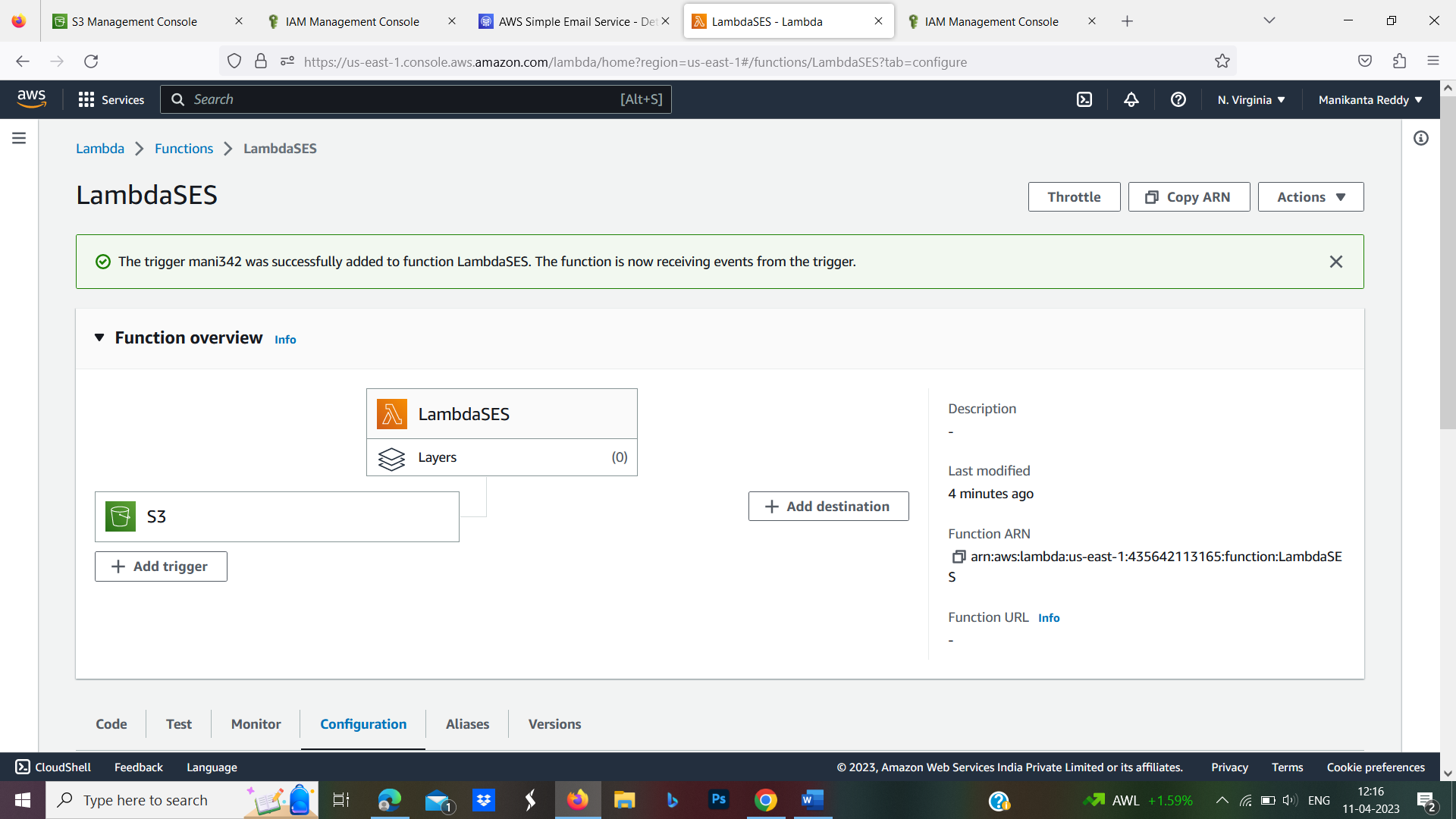
Step-20: Now scroll back up and click on **Add Trigger.**Select ‘S3’ as the trigger, the bucket name which we have created at the beginning of the exercise, select the ‘Event type’ to ‘PUT’ and finally add prefix & suffix (if any) for your bucket.



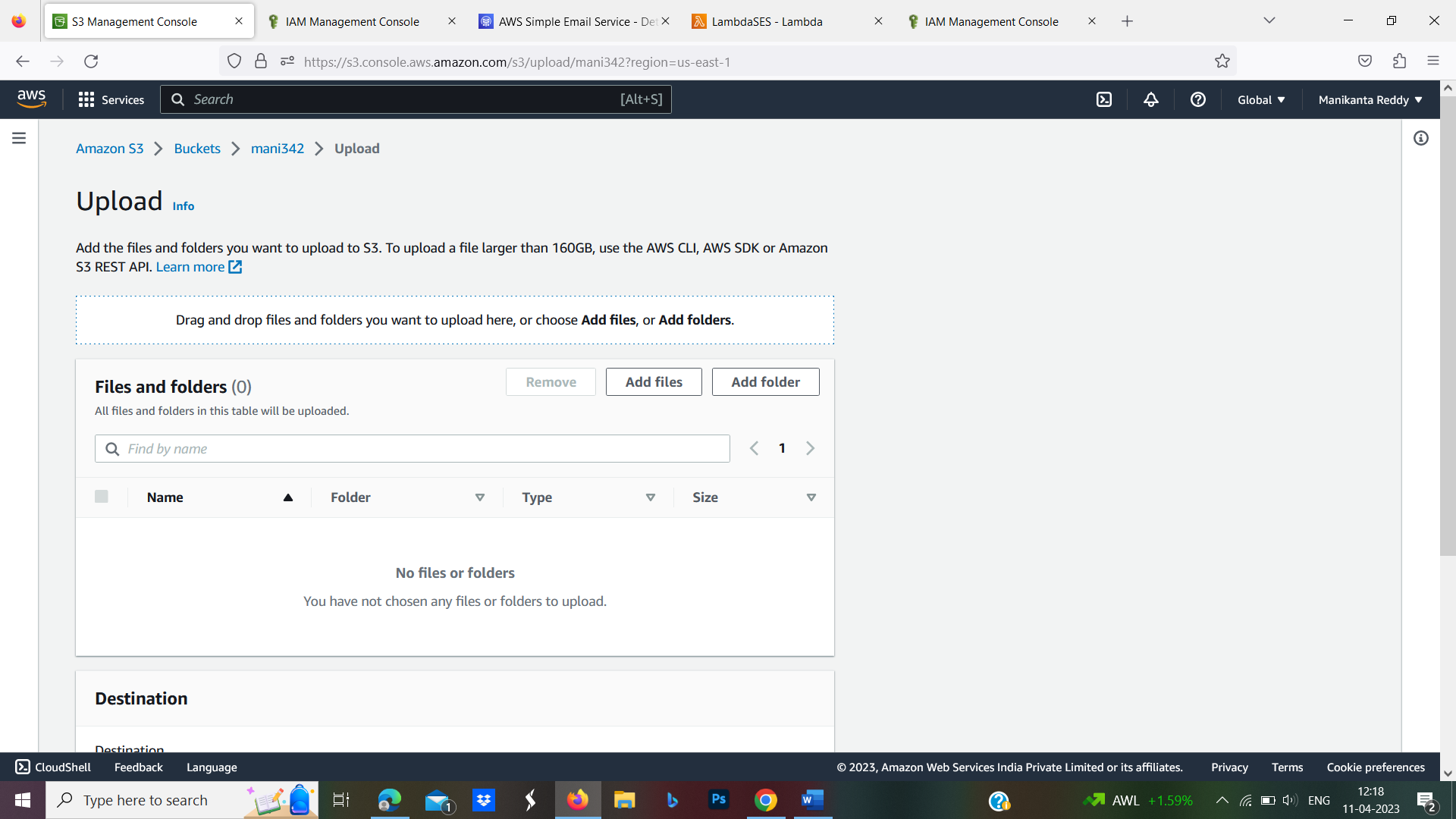
Step-21: Scroll down and click on Add

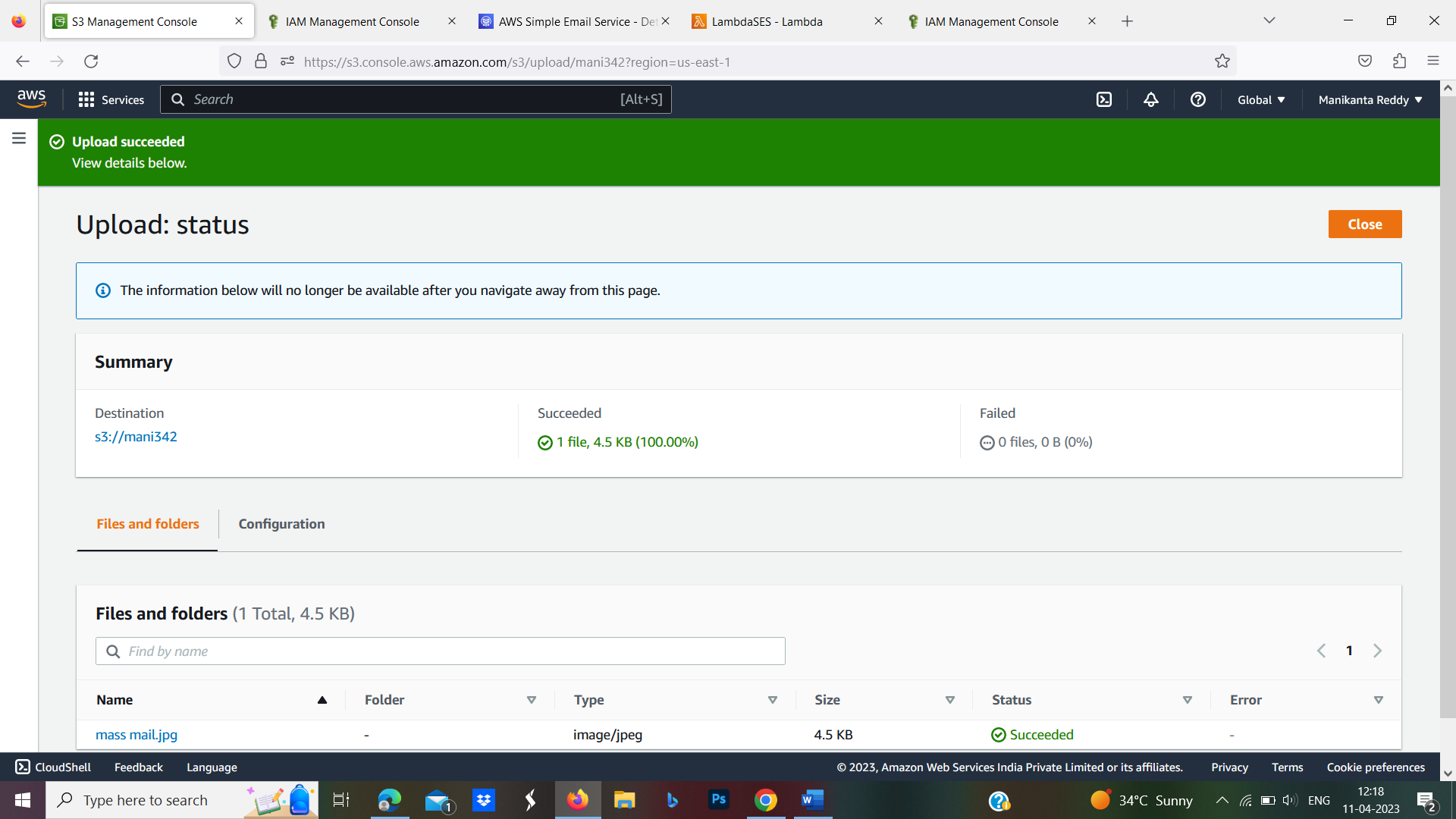


Step-22: We can see the S3 trigger added on the lambda console.

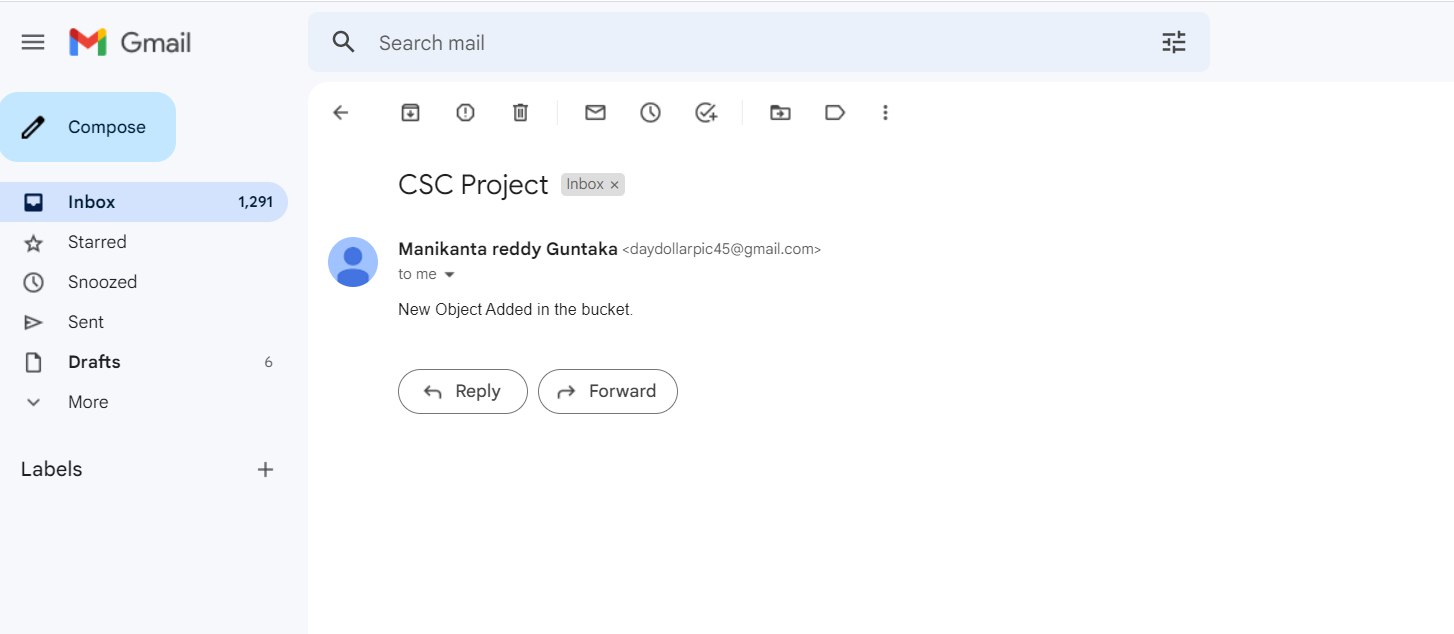


Step-23: Now, go back to the Amazon S3 console and get into the bucket which we have created in the beginning of the exercise.





Step-24: As soon as the image object is added to the bucket, it will trigger the Lambda function and the admin will receive an email on its email address



## **Conclusion:**

* In this Documentation, you have seen how easily we can configure Amazon Simple Email Service and we have used it to send notifications while any new objects get added to the S3 bucket. We have used the Lambda function to write code and configured triggers.

**THANK YOU……….**