AWS SES Configuration

In this tutorial we will configure the components required by Amazon SES (Simple Email Service) to allow us to send outbound emails from an EC2 instance.

Prerequisites

- an AWS Free Tier account
- AWS Ubuntu 20 EC2 instance
- AWS RHEL 8 EC2 instance
- an email address
- internet access

If you do not have an AWS account, you can access my **AWS Create Free Tier Account** tutorial here.

If you do not have an AWS Ubuntu 20 EC2 instance, my tutorial **Create AWS Ubuntu 20 EC2 Instance** is here. If you do not have an AWS RHEL 8 EC2 instance, my tutorial **Create AWS RHEL 8 EC2 Instance** is here.

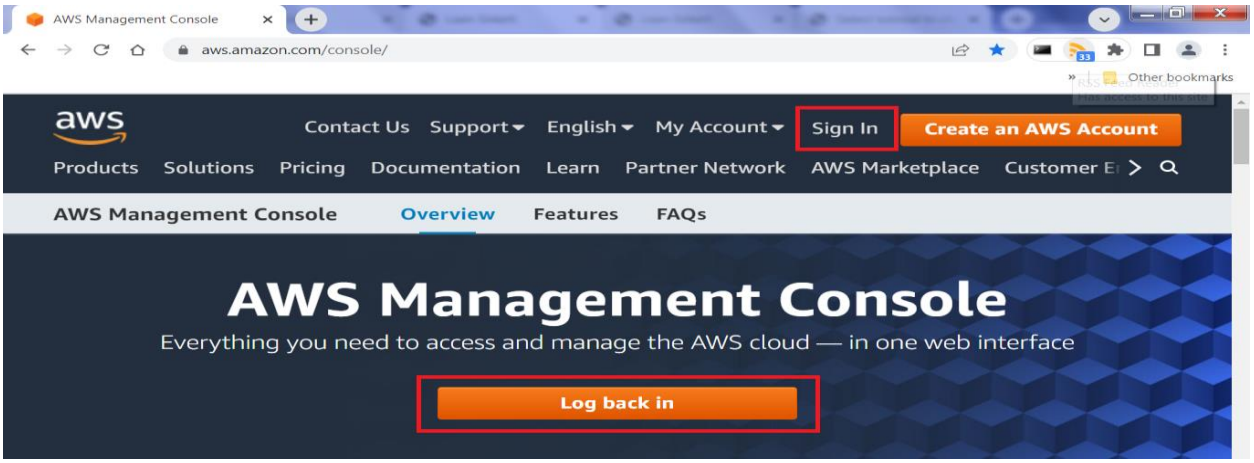
After completing this tutorial, you will be ready for my Postfix tutorials where I demonstrate the installation, and configuration, of Postfix as an outbound send-only email server, on an Ubuntu 20 EC2 & RHEL 8 EC2. Postfix will allow you to fully utilize Amazon SES.

The **AWS Ubuntu 20 EC2 Postfix Install** tutorial is accessible **here**, while the **AWS RHEL 8 EC2 Postfix Install** tutorial is accessible **here**.

Steps to complete tutorial:

- Gather EC2 Instance Information
- Change EC2 Instance Security Group
- Add Rule to Security Group
- Create Verified Identity
- Create SMTP Credentials
- Create VPC Endpoint
- Simple Email Service Test

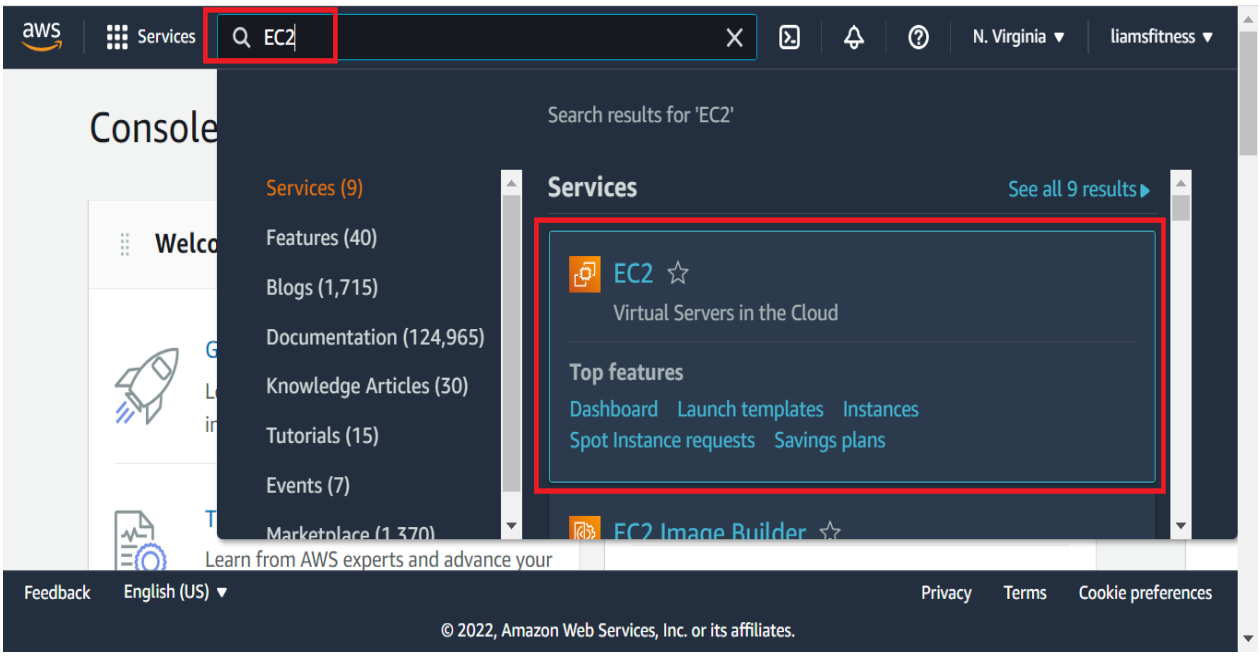
To begin, go to the following website, https://aws.amazon.com/console/ and log in to the console.



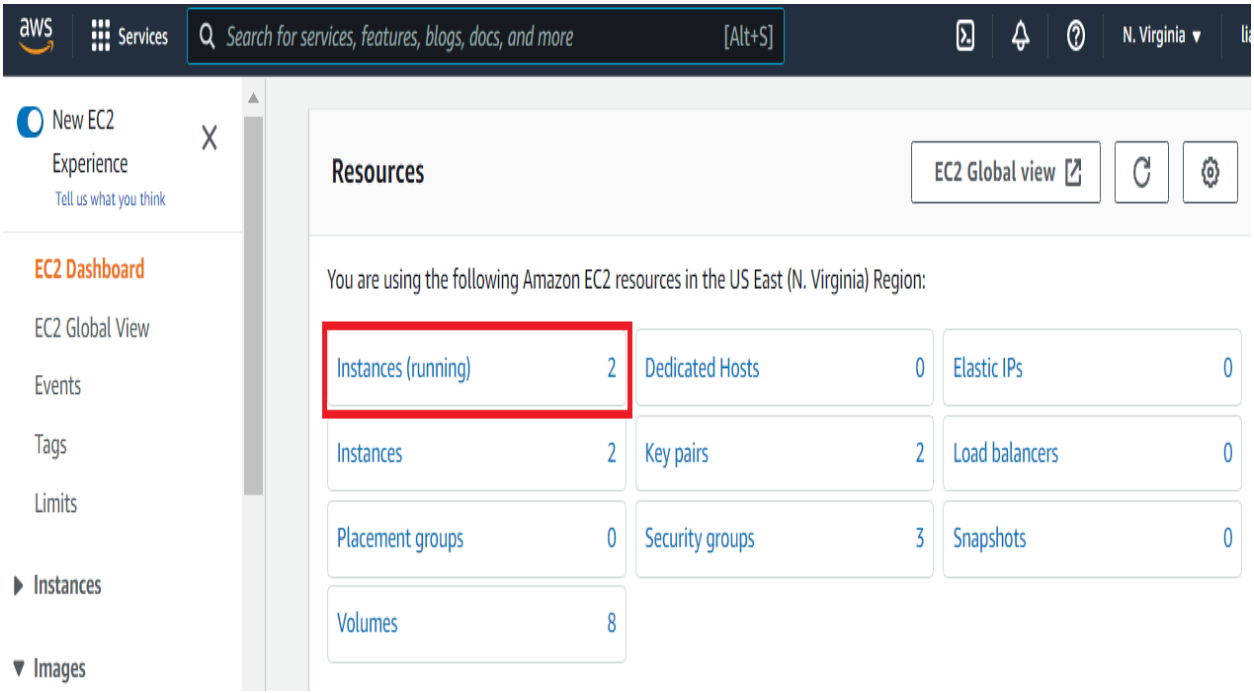
Before we configure the Amazon Simple Email Service, I will first gather the necessary information from both my EC2 instances: **security group**, **subnet** & **availability zone**. All of these values will be used during the VPC Endpoint creation, while the subnet will also be used when adding a new inbound rule to the security group.

Gather EC2 Instance Information

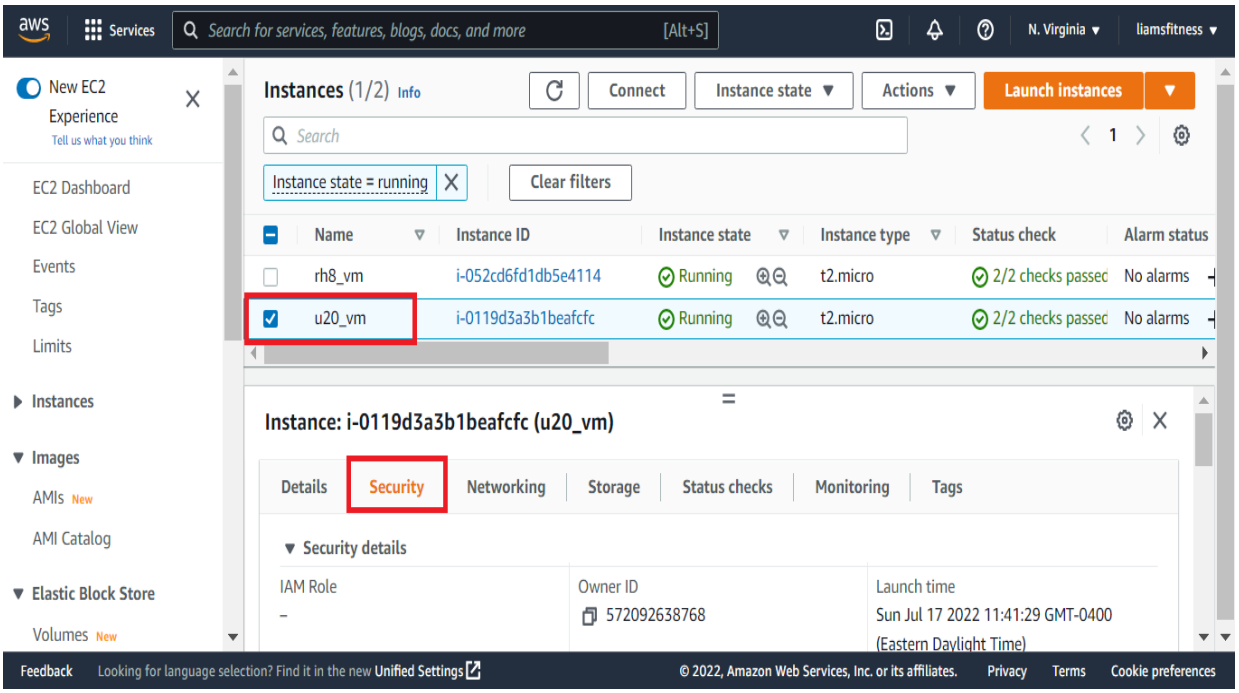
Once logged in, enter **EC2** in the search bar and select **EC2** Virtual Servers in the Cloud.



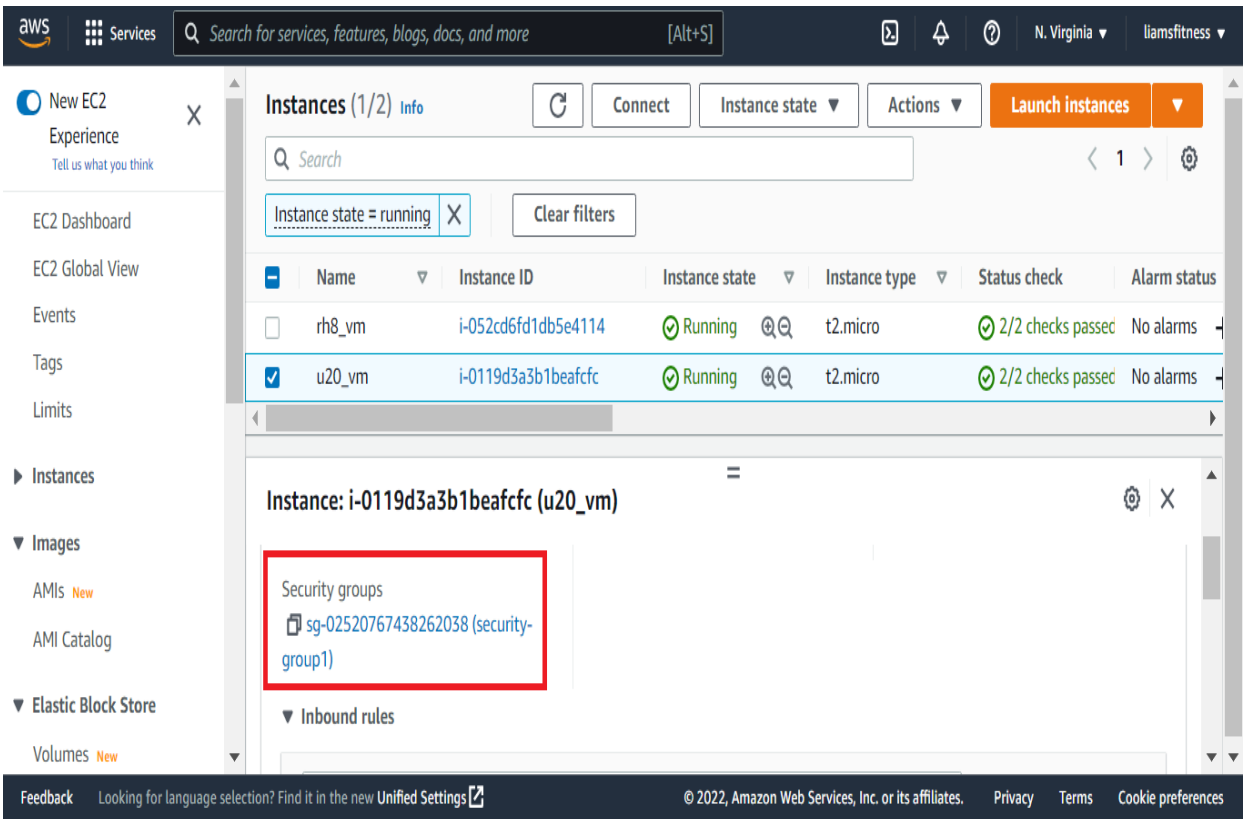
On the EC2 Dashboard, select **Instances (running)**



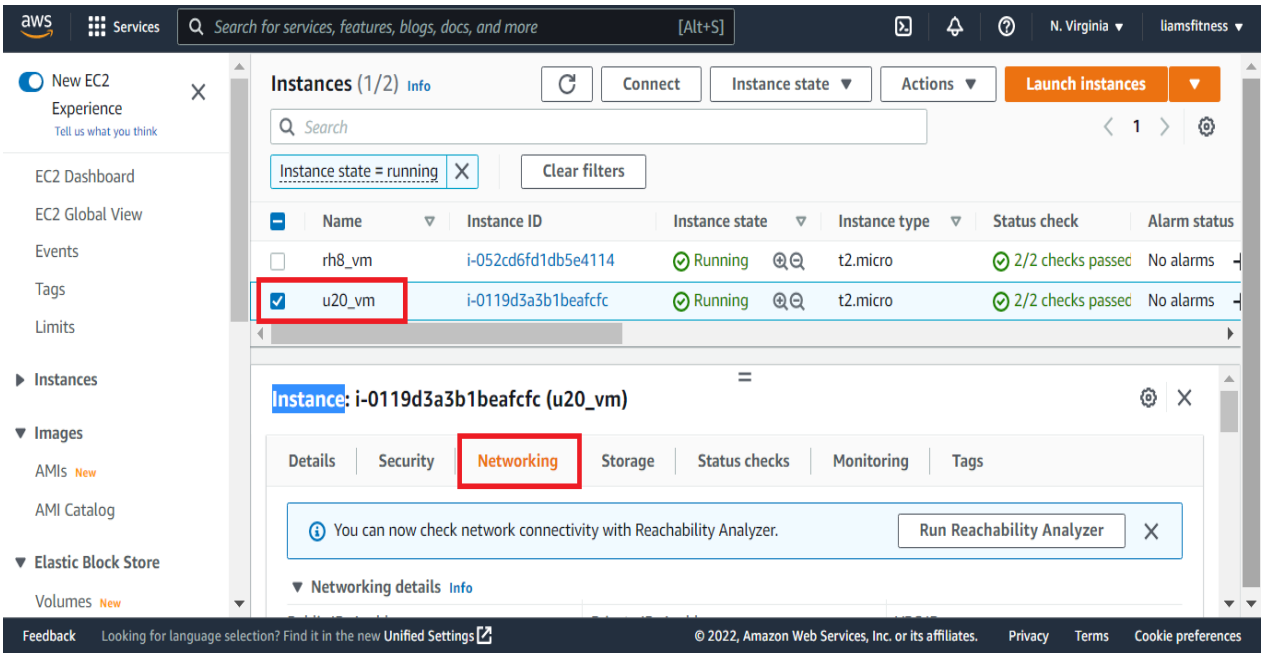
On the Instances screen, ensure an EC2 instance is selected and that the **Security** tab is selected. I will start with my Ubuntu 20 EC2 instance (**u20_vm**).



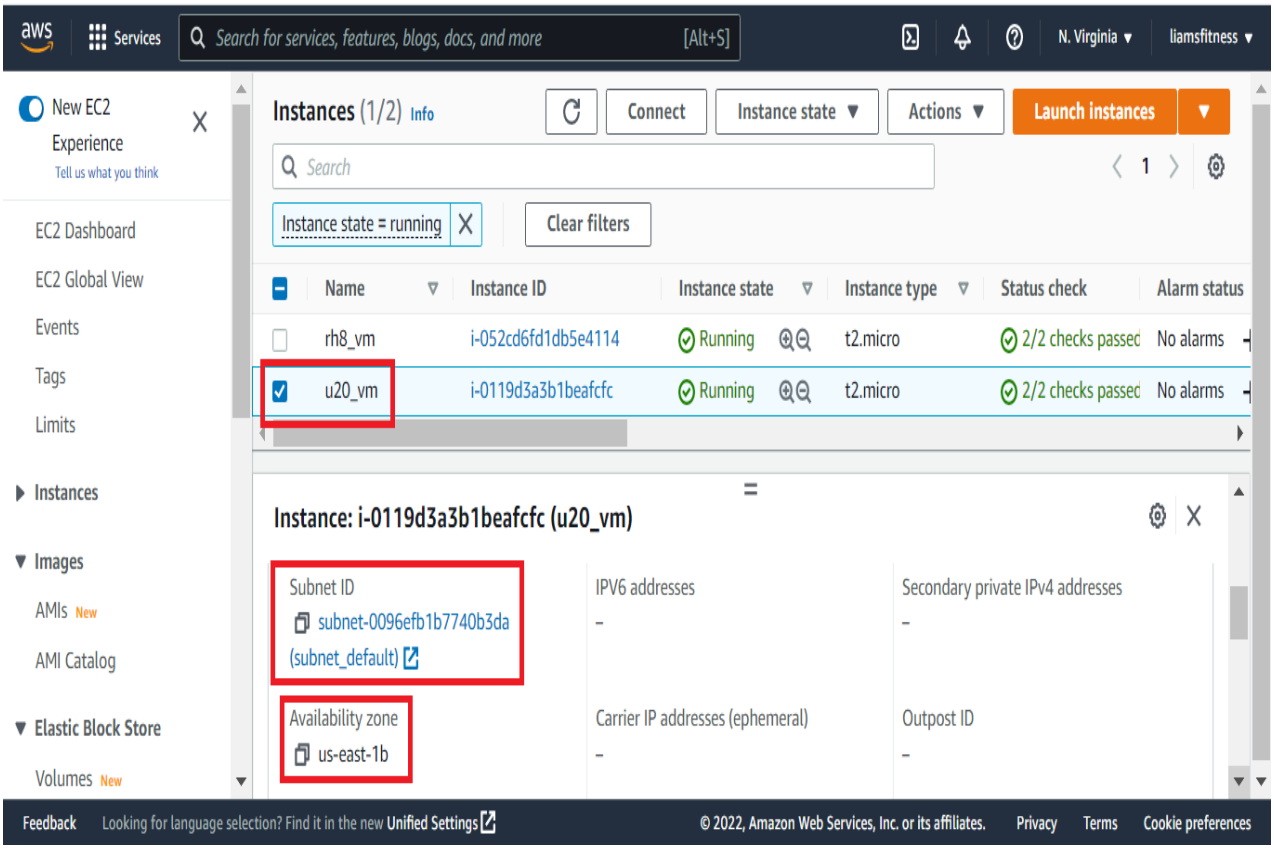
In the bottom section of the screen, scroll down until **Security groups** is visible and note the value. In my case the value is **security-group1**.



Next, in the bottom of the screen, scroll back up until the tabbed menu is visible and select the **Networking** tab.

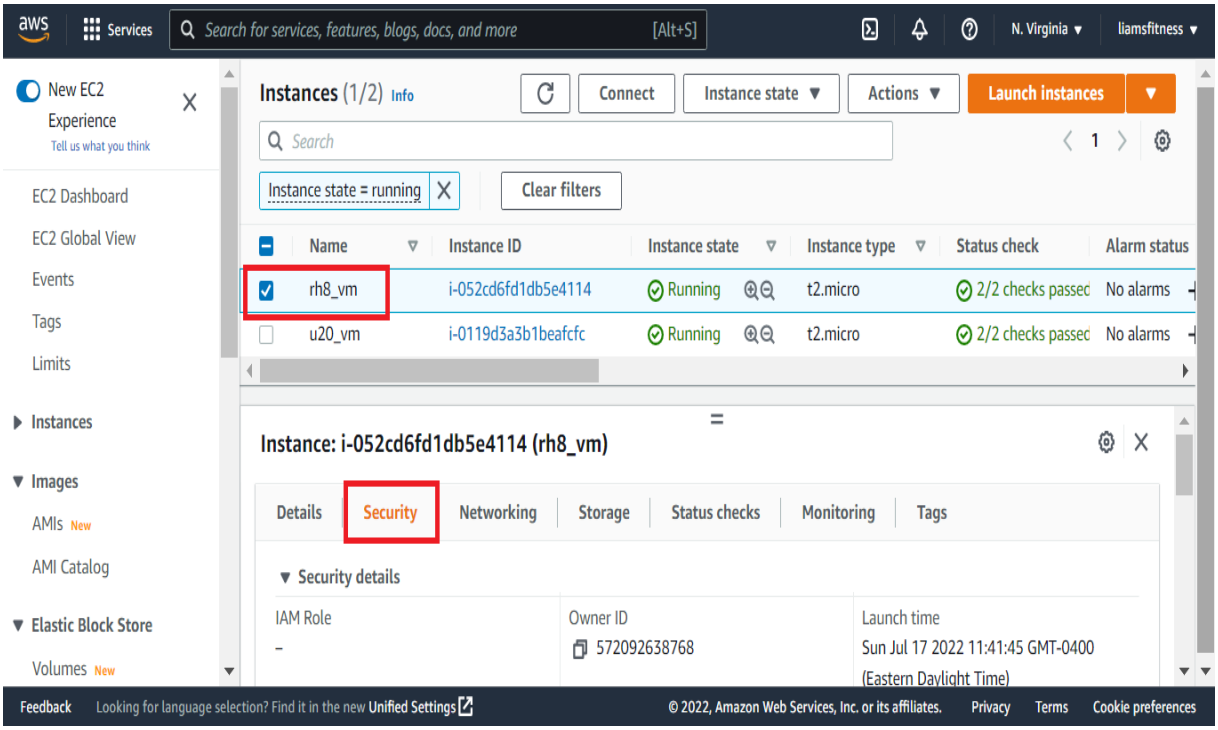


In the bottom section of the screen, under the **Networking** tab, scroll down until **Subnet ID** and **Availability zone** are visible and note the values. In my case the values are **subnet_default** & **us-east-1b**

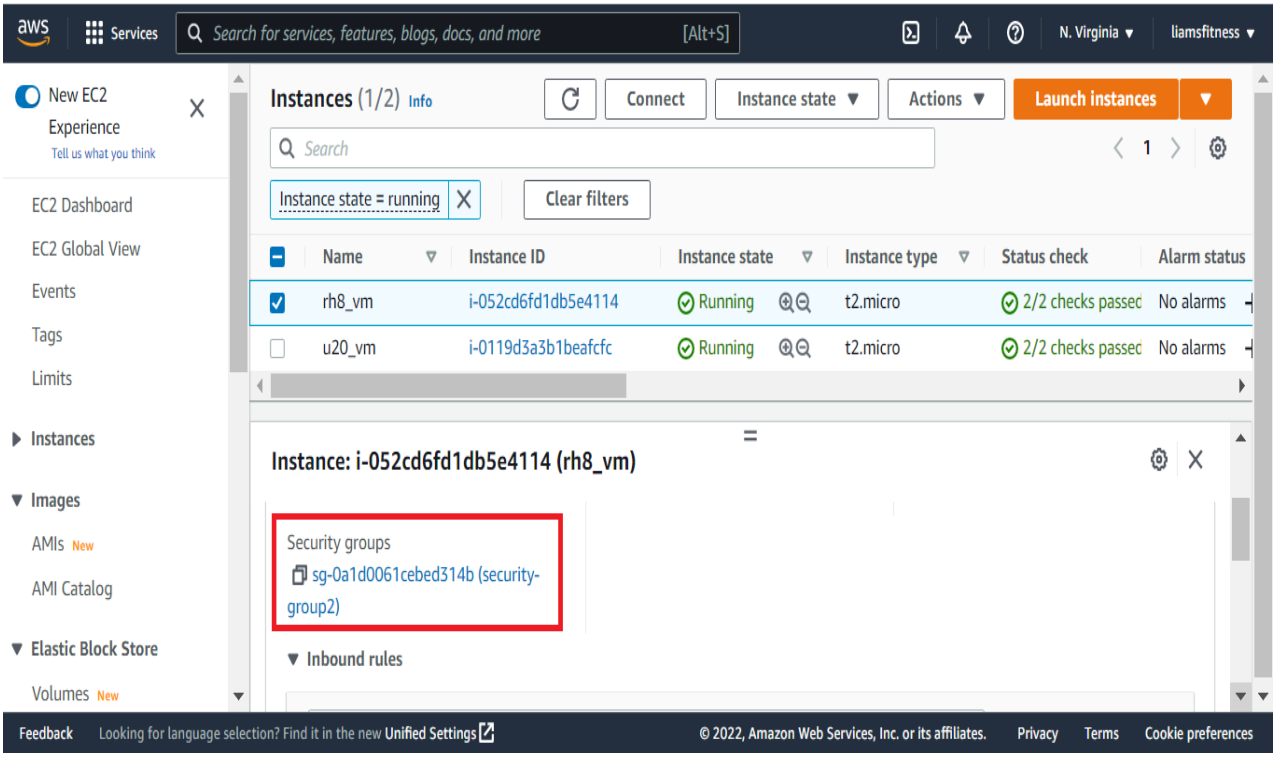


Now I will perform the same steps to gather information on my RHEL 8 EC2 instance (**rh8_vm**).

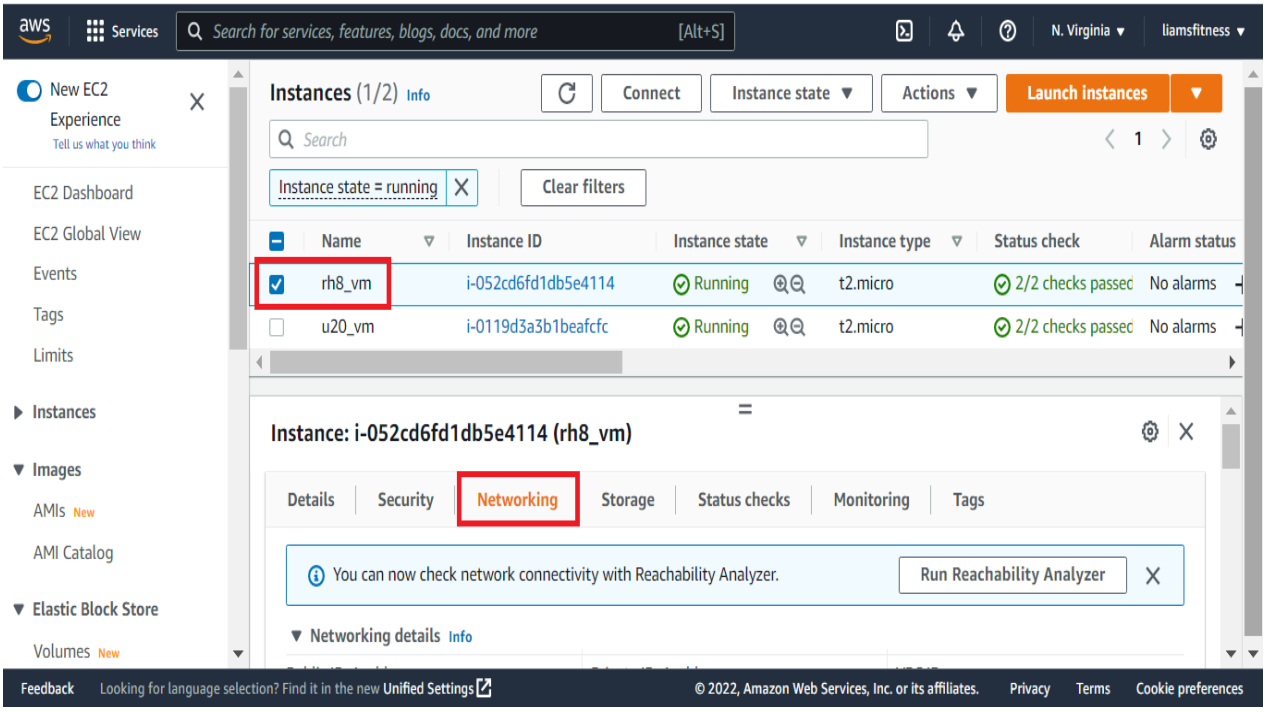
On the Instances screen, I will select **rh8_vm** and ensure that the **Security** tab is selected.



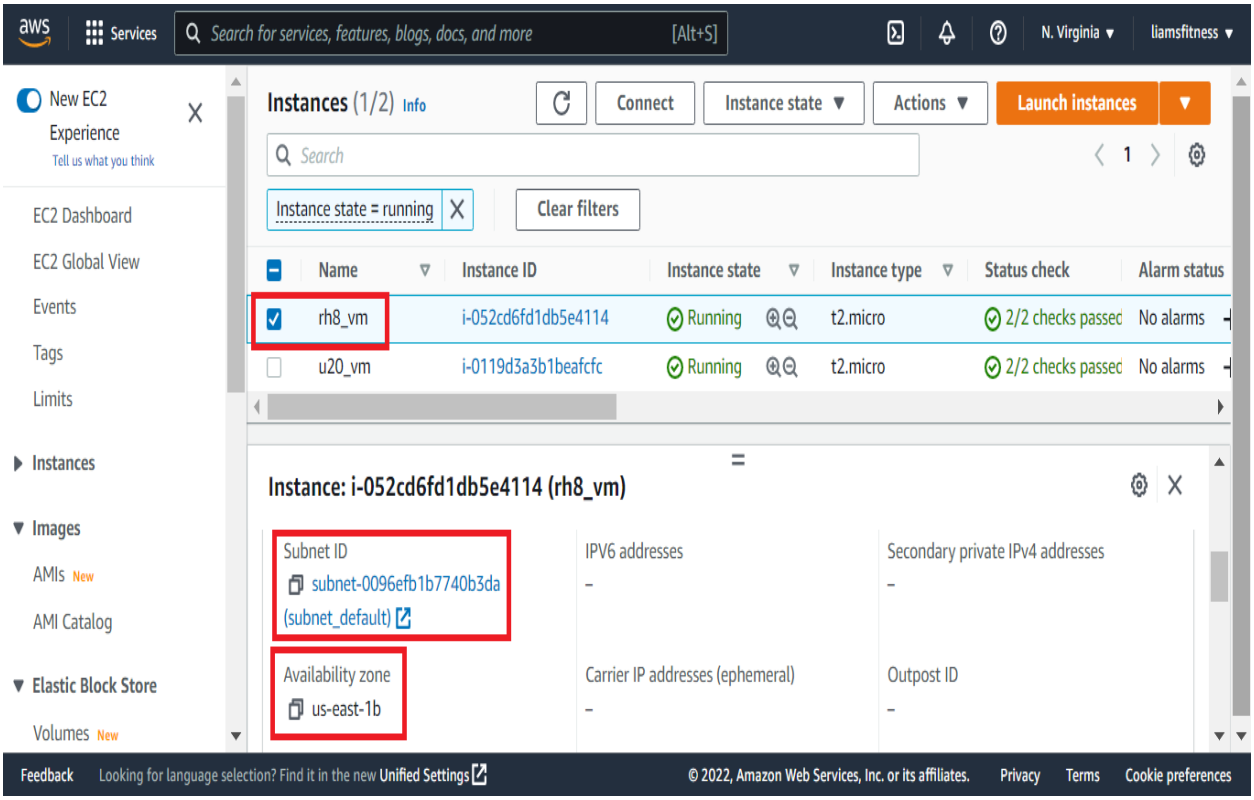
In the bottom section of the screen, scroll down until **Security groups** is visible and note the value. In my case the value is **security-group2**.



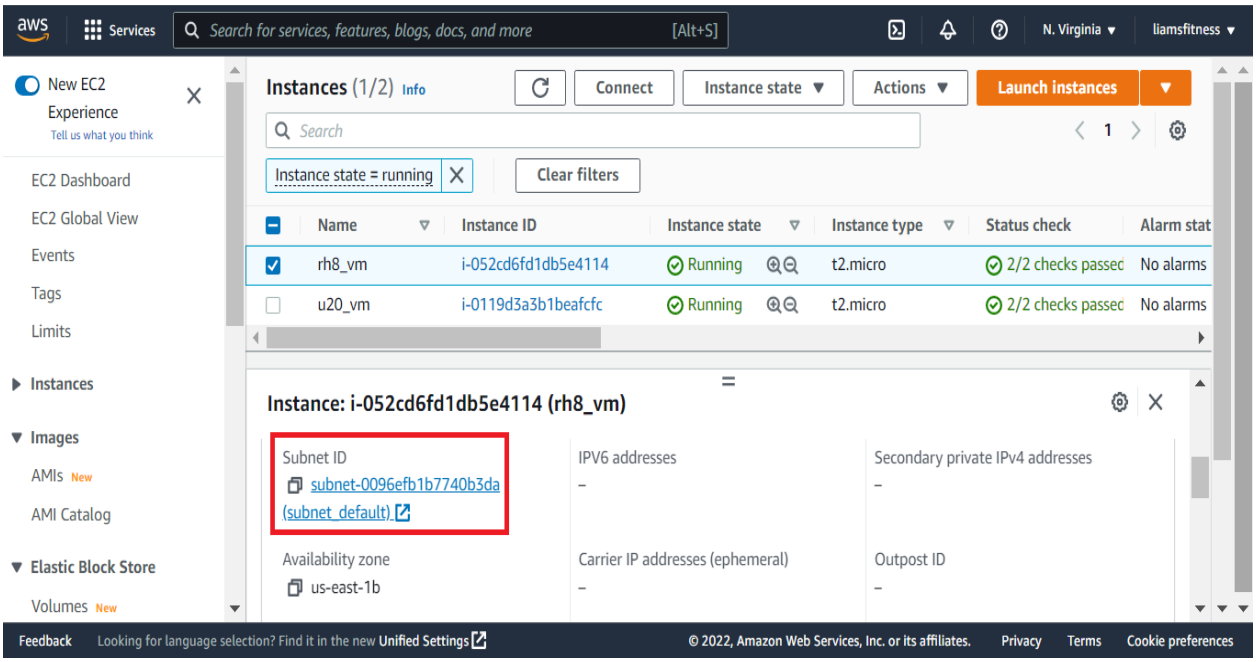
Next, in the bottom of the screen, scroll back up until the tabbed menu is visible and select the **Networking** tab.



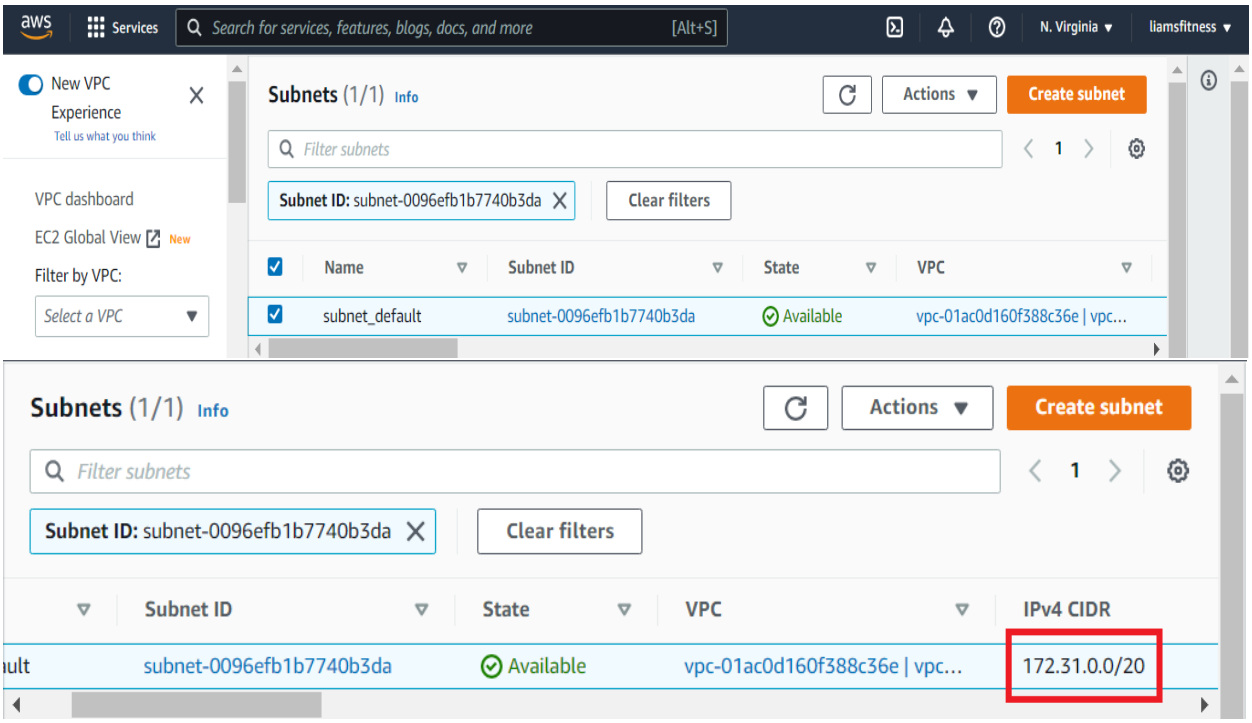
In the bottom section of the screen, under the **Networking** tab, scroll down until **Subnet ID** and **Availability zone** are visible and note the values. In my case the values are **subnet_default** & **us-east-1b**.



After gathering the necessary EC2 instance information, we need to open a port (587) through the firewall by adding an inbound rule to the security group (**security-group1**). One of the required rule parameters is the **Source** field (*from where these requests will be coming from*). The value I will be using is the CIDR IPv4 subnet of both my EC2 instances. To access this value, click the link under the **Subnet ID**



The **Subnets** screen will open in a new tab. On the **Subnets** screen, scroll to the right until the **IPv4 CIDR** heading is visible and note the value.



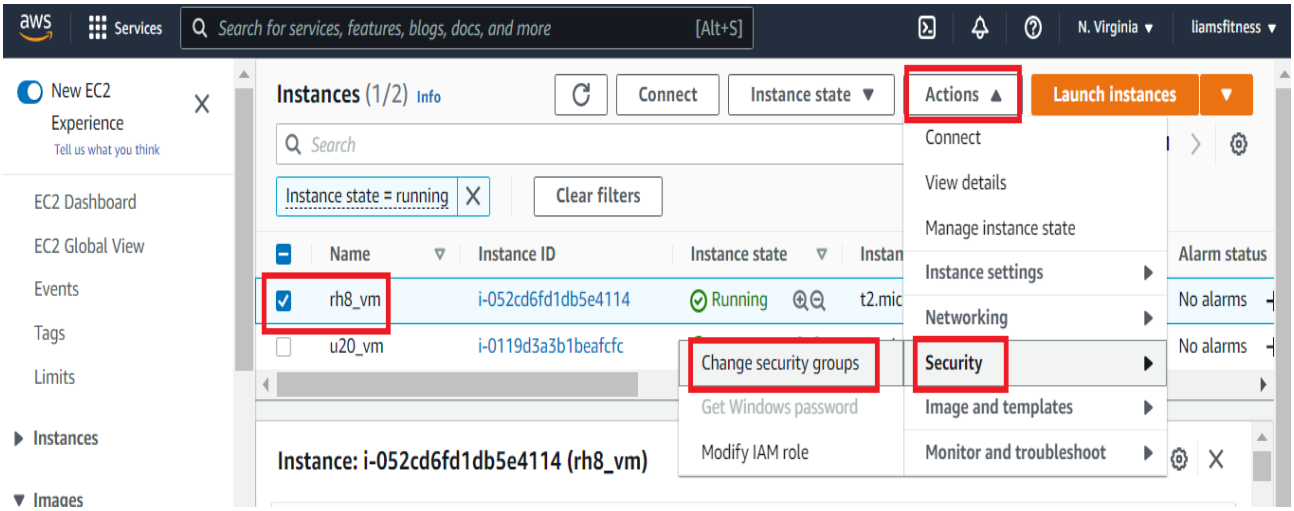
In my case, the **IPv4 CIDR** value is **172.31.0.0/20**.

At the end of this tutorial, I want both of my EC2 instances to be able to send outbound emails. Normally, port 25 is used to send email. Unfortunately, Amazon prevents the use of port 25. Instead, we will use port 587. Therefore, we will need to add an inbound rule for port 587 to the security group (**security-group1**). Since my RHEL 8 EC2 instance is using **security-group2**, I will first change its security group to **security-group1**. This will prevent me from having to add separate rules to different security groups.

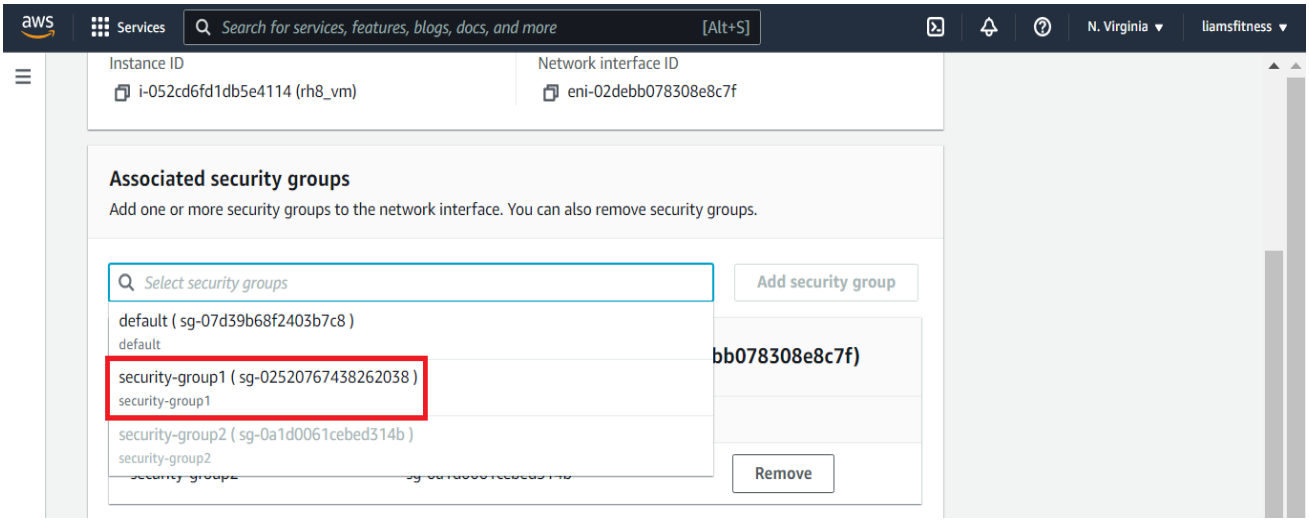
After making that change and adding a new inbound rule to **security-group1**, both EC2 instances will be capable of sending outbound emails.

Change EC2 Instance Security Group

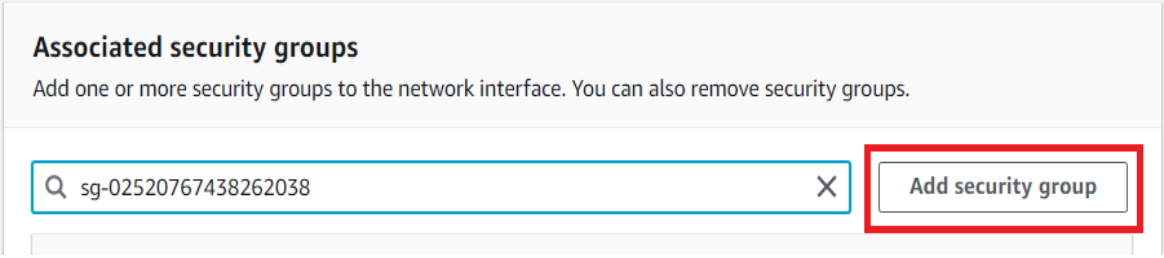
Back on the **Instances** screen, to change my RHEL 8 EC2 instance's security group, I will first ensure it is selected. Then, I will click the **Actions** menu, scroll down and click **Security** followed by **change security groups**.



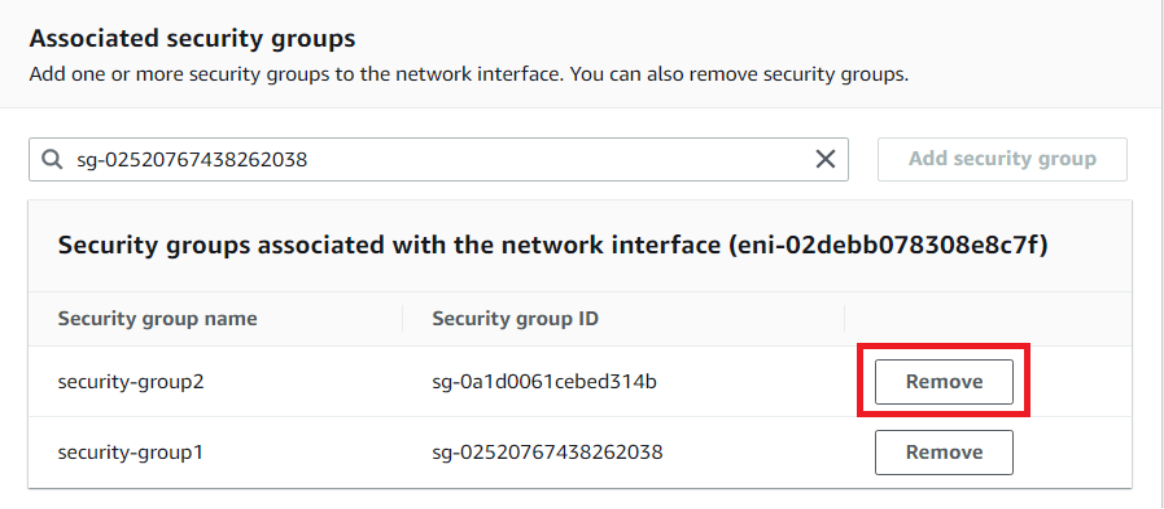
On the **Change security groups** screen, scroll down to the Associated security groups section and place your cursor in the greyed out ***Select security groups*** text box. A listing of available security groups will appear. I have selected **security-group1**



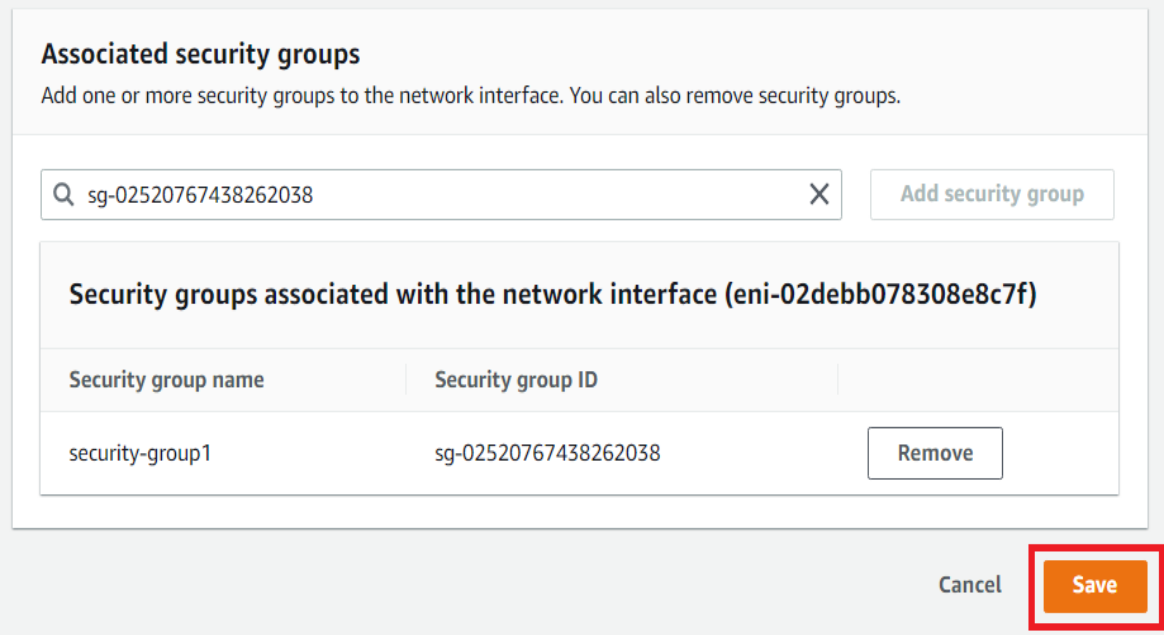
Next, click **Add security group**

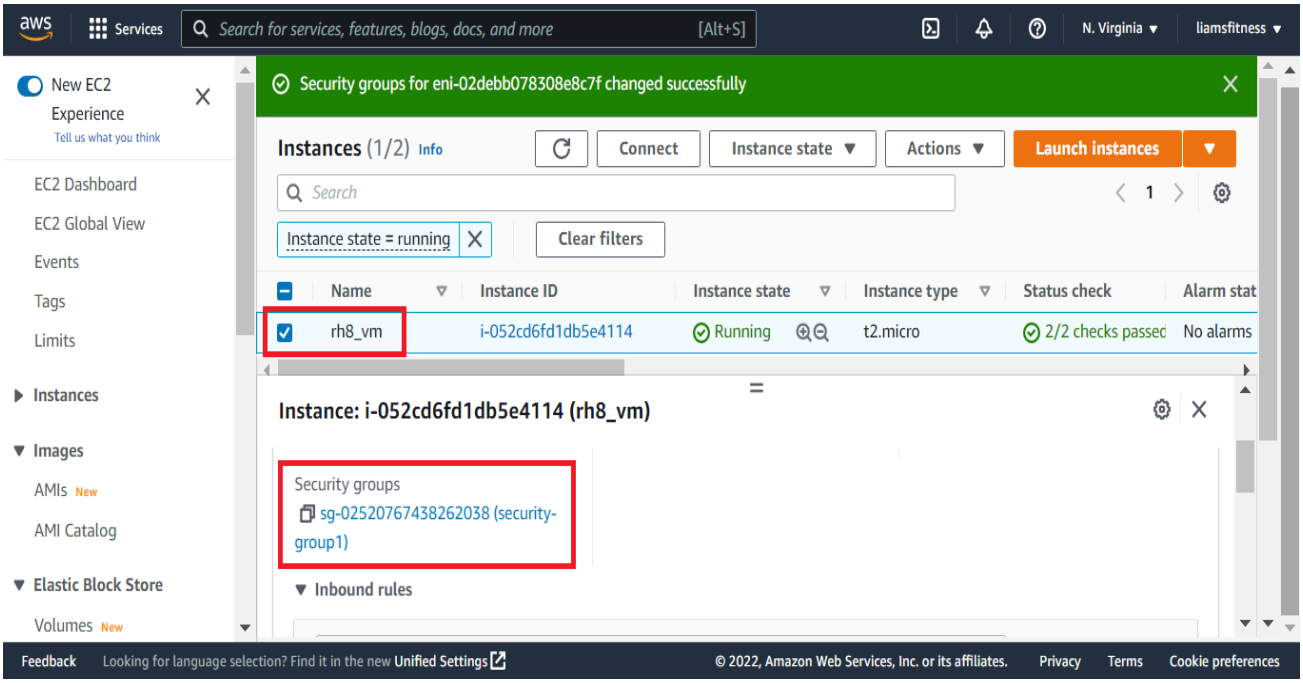


Then, next to the original security group (**security-group2**) click **Remove**



Finally, to save the changes, click **Save**



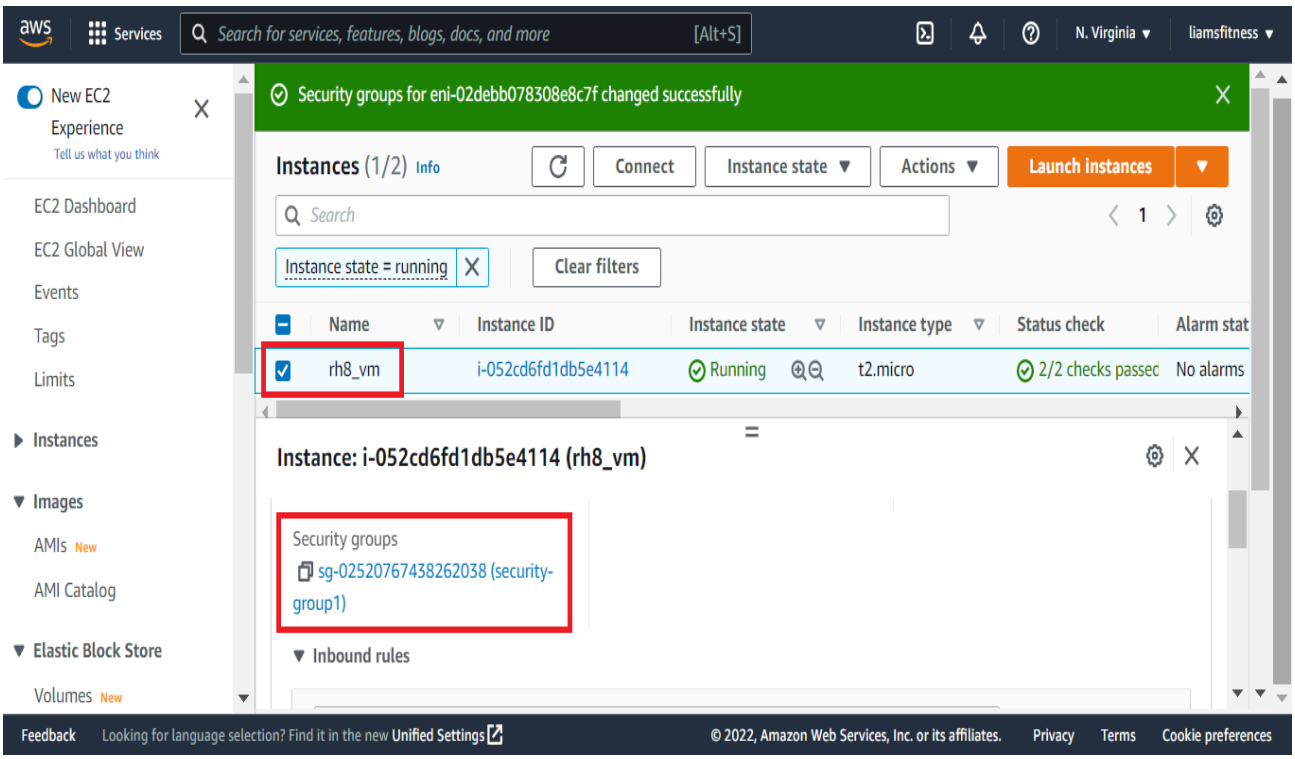


The security group is now the same for both of my EC2 instances: **security-group1**

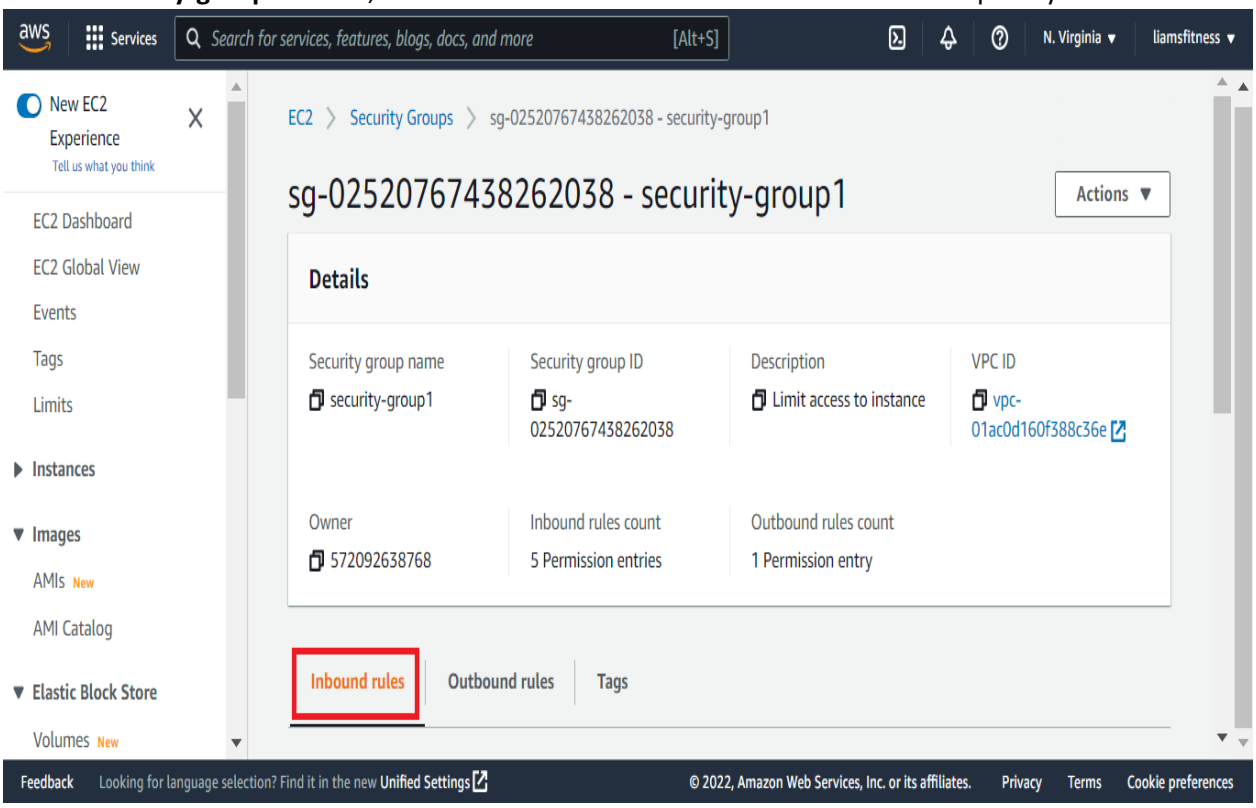
The next step is to add an inbound rule for port 587 to **security-group1**.

Add Rule to Security Group

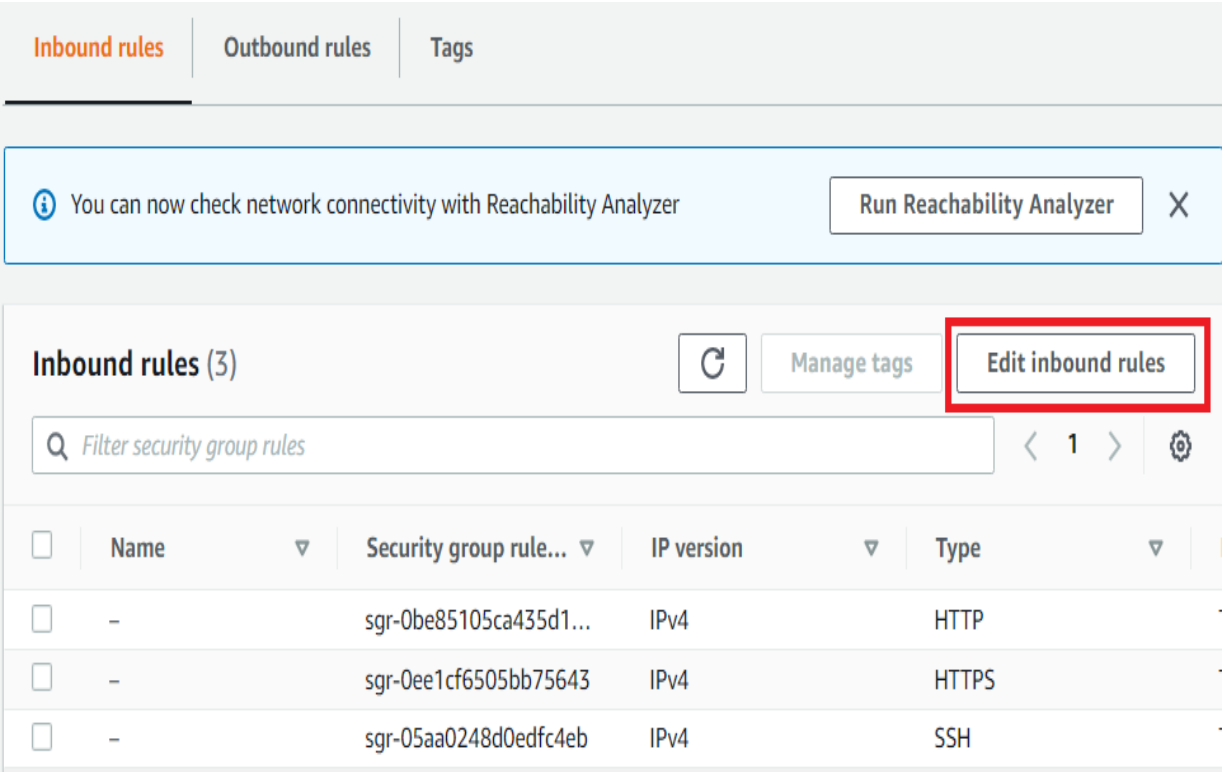
To access the **Security Groups** screen, at the bottom of the **Instances** screen, under the **Security** tab, locate **Security groups** and click the security group link (in my case, **security-group1**).



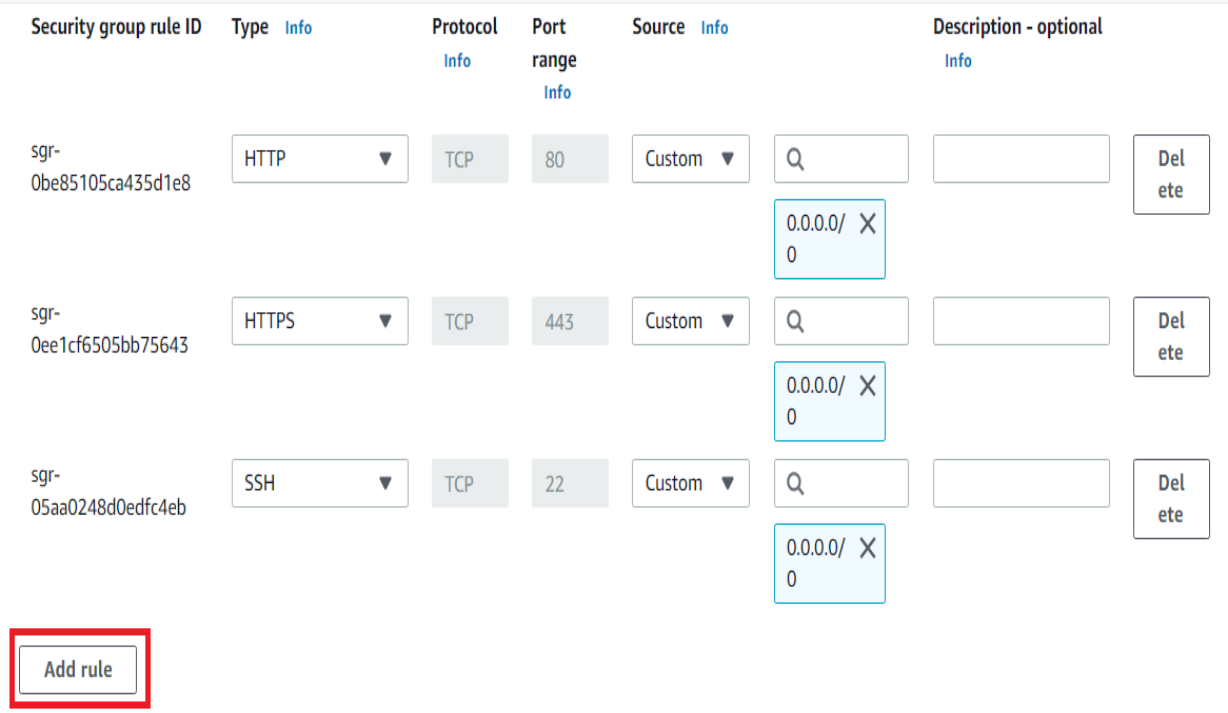
On the **security-group1** screen, scroll down until the **Inbound rules** section is completely visible.



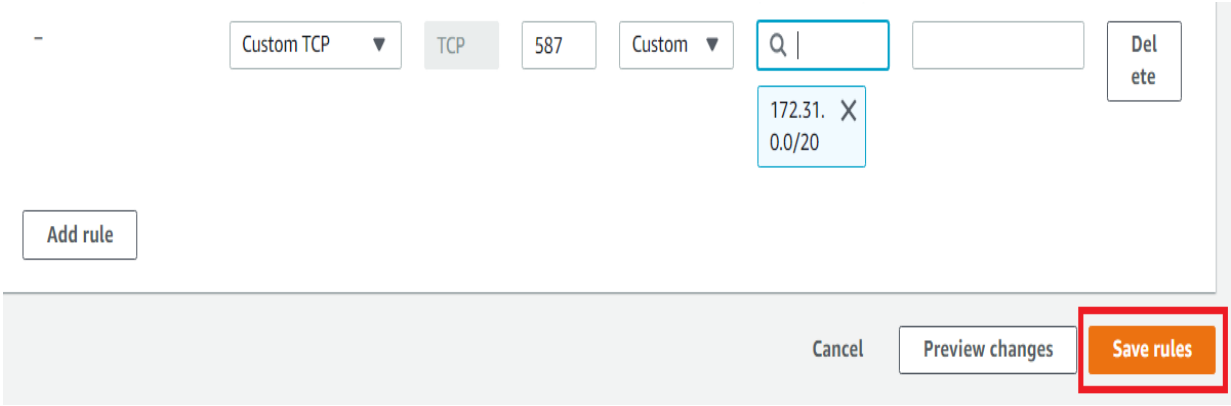
Next, click **Edit Inbound Rules**

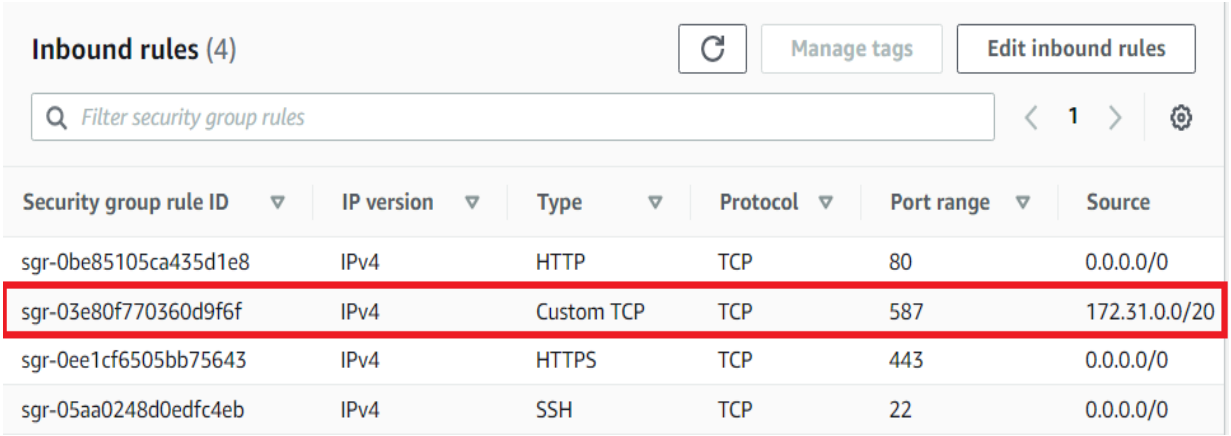


On the **Edit Inbound Rules** screen, click **Add rule**



Ensure the **Port range** is set to **587** and **Source** is set to **172.31.0.0/20** and click **Save rules**.

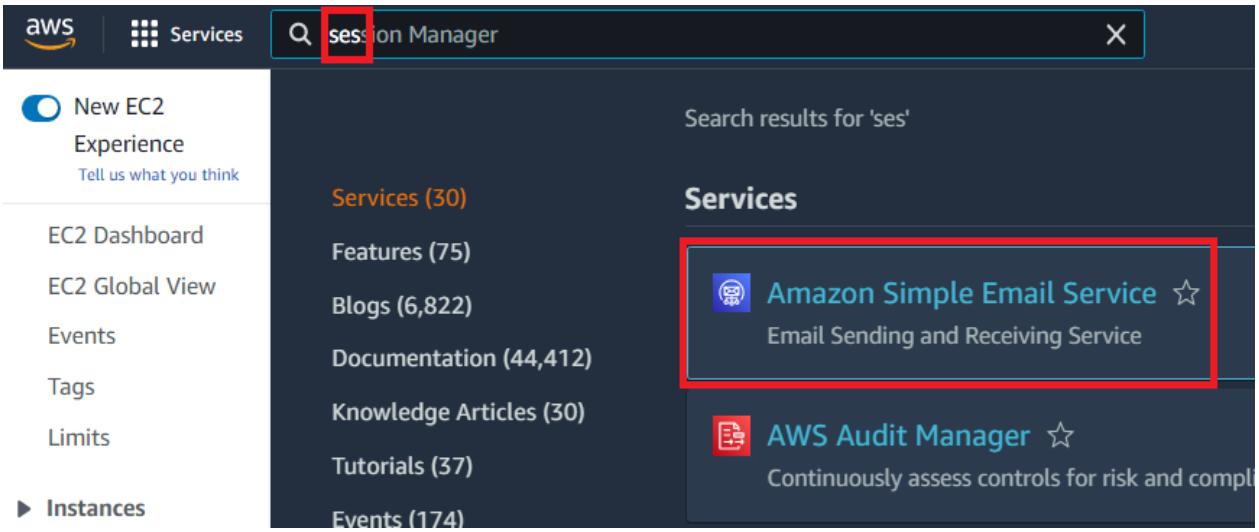




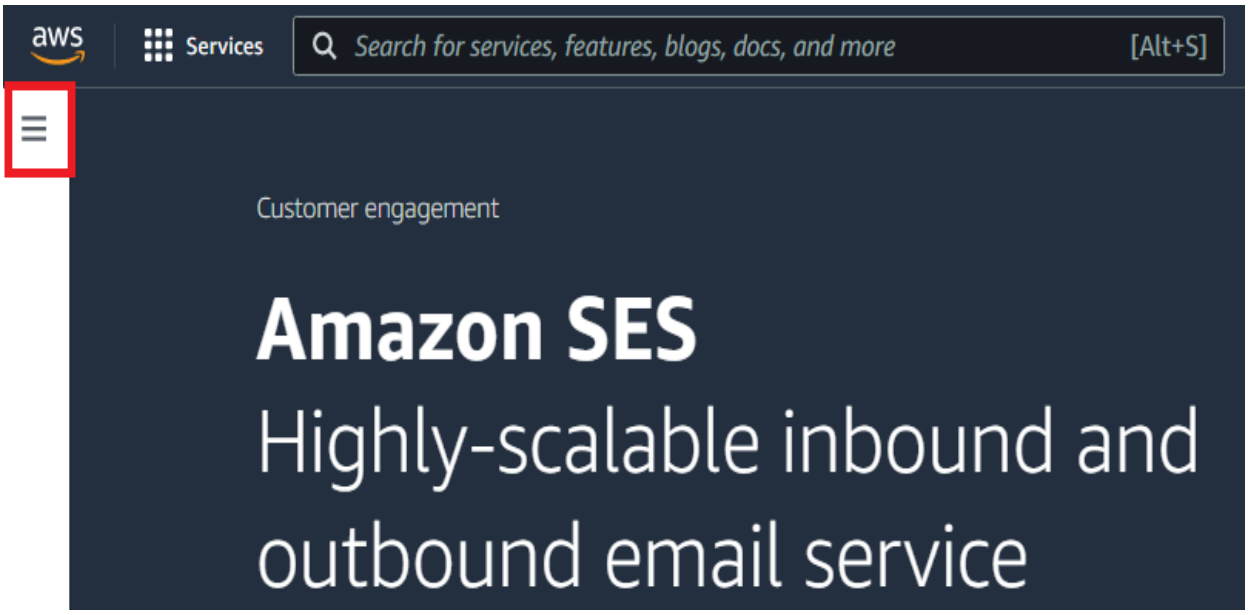
Create Verified Identity

In order to use the Amazon's SES (Simple Email Service), the first step is to create a verified identity which can be either a domain or an email address. I will be creating a verified identity using an email address.

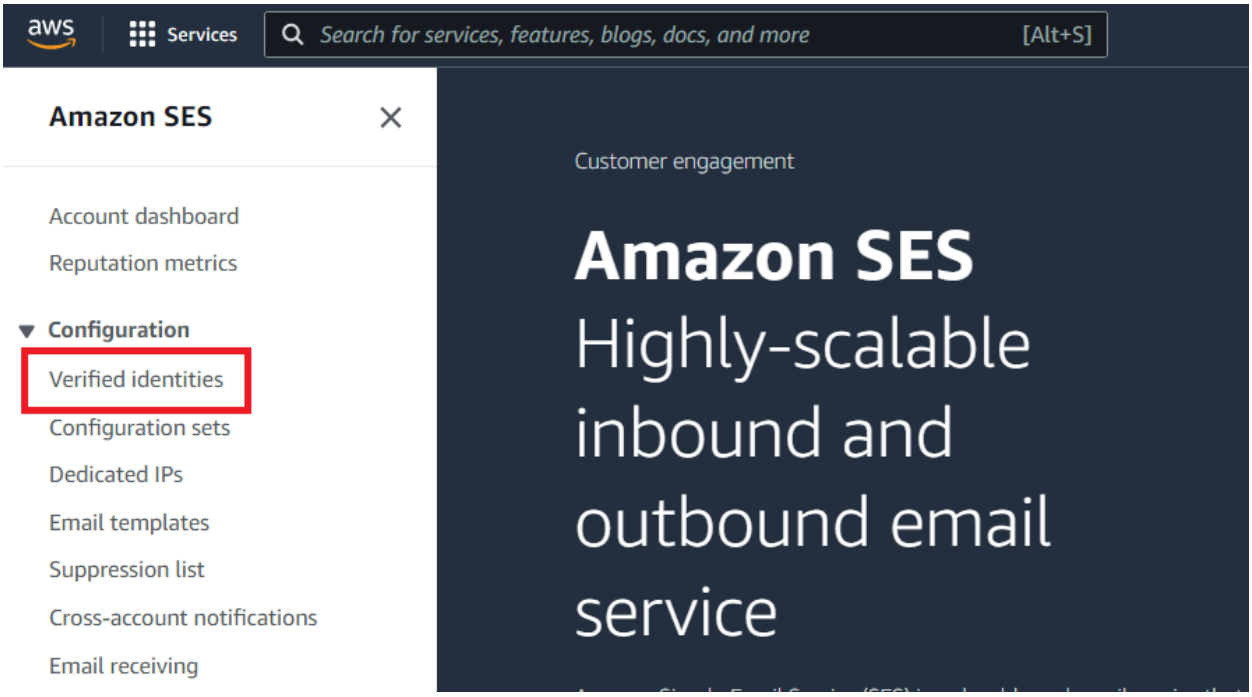
To access the SES service, at the top of the screen, type **ses** in the search bar and select **Amazon Simple Email Service**.



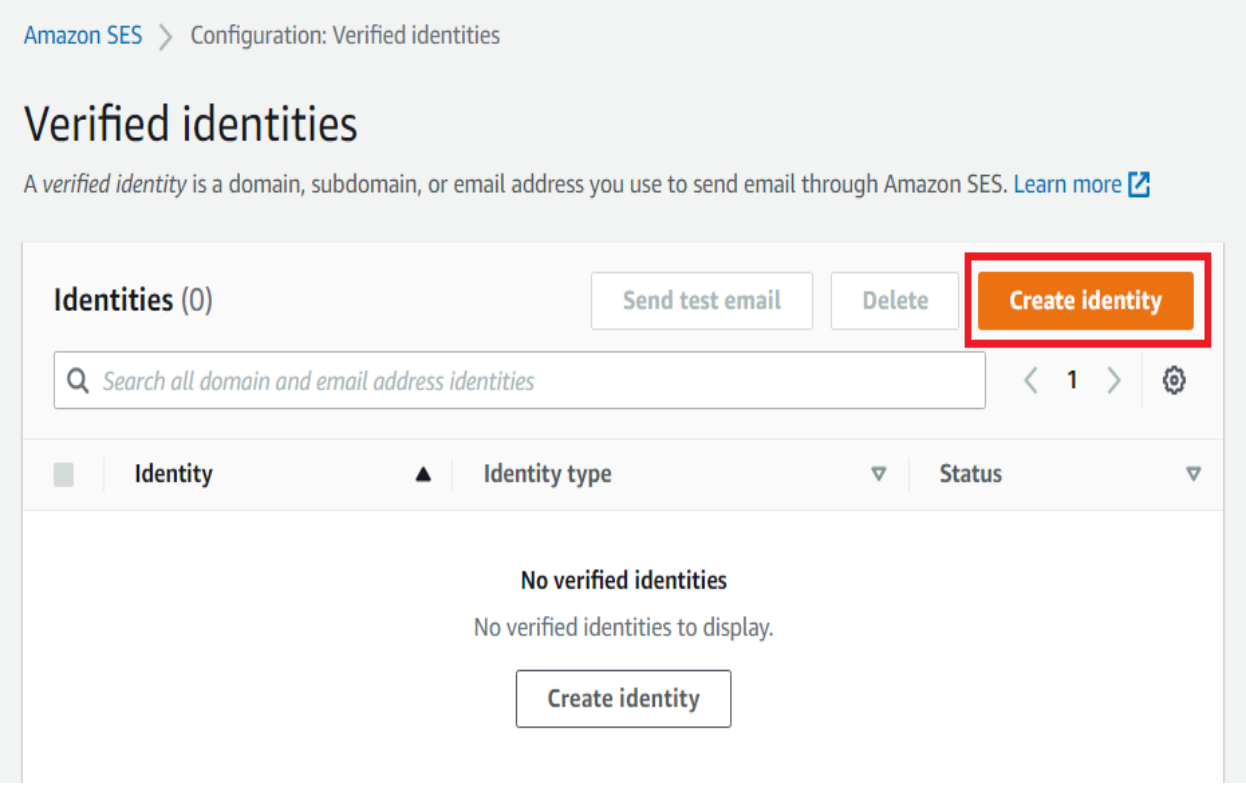
On the SES screen, click the accordion menu in the top left of the screen to access the available options.



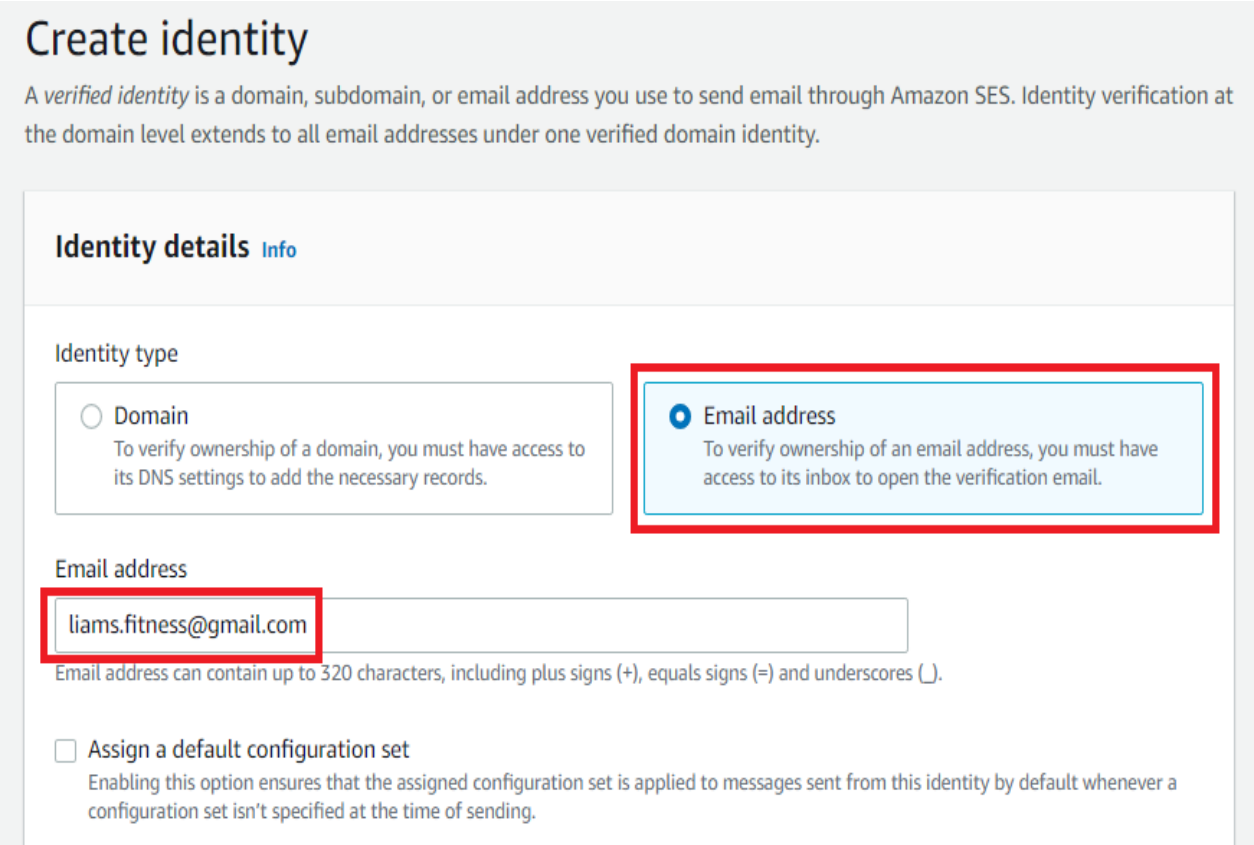
Under **Configuration**, click **Verified Identities**



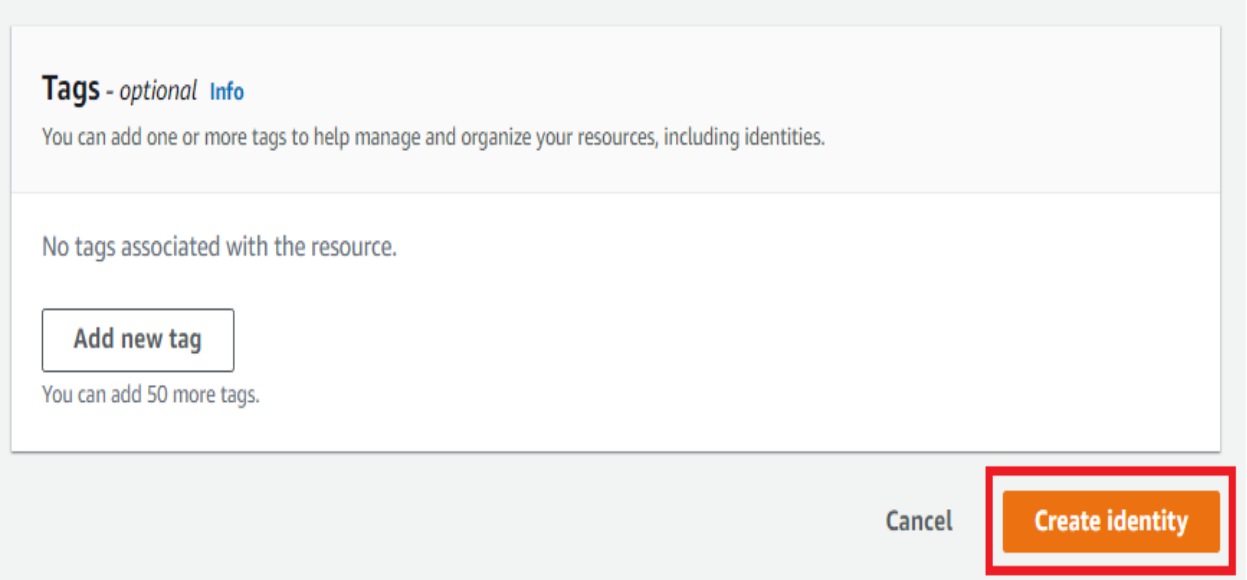
On the **Verified identities screen**, click **Create identity**



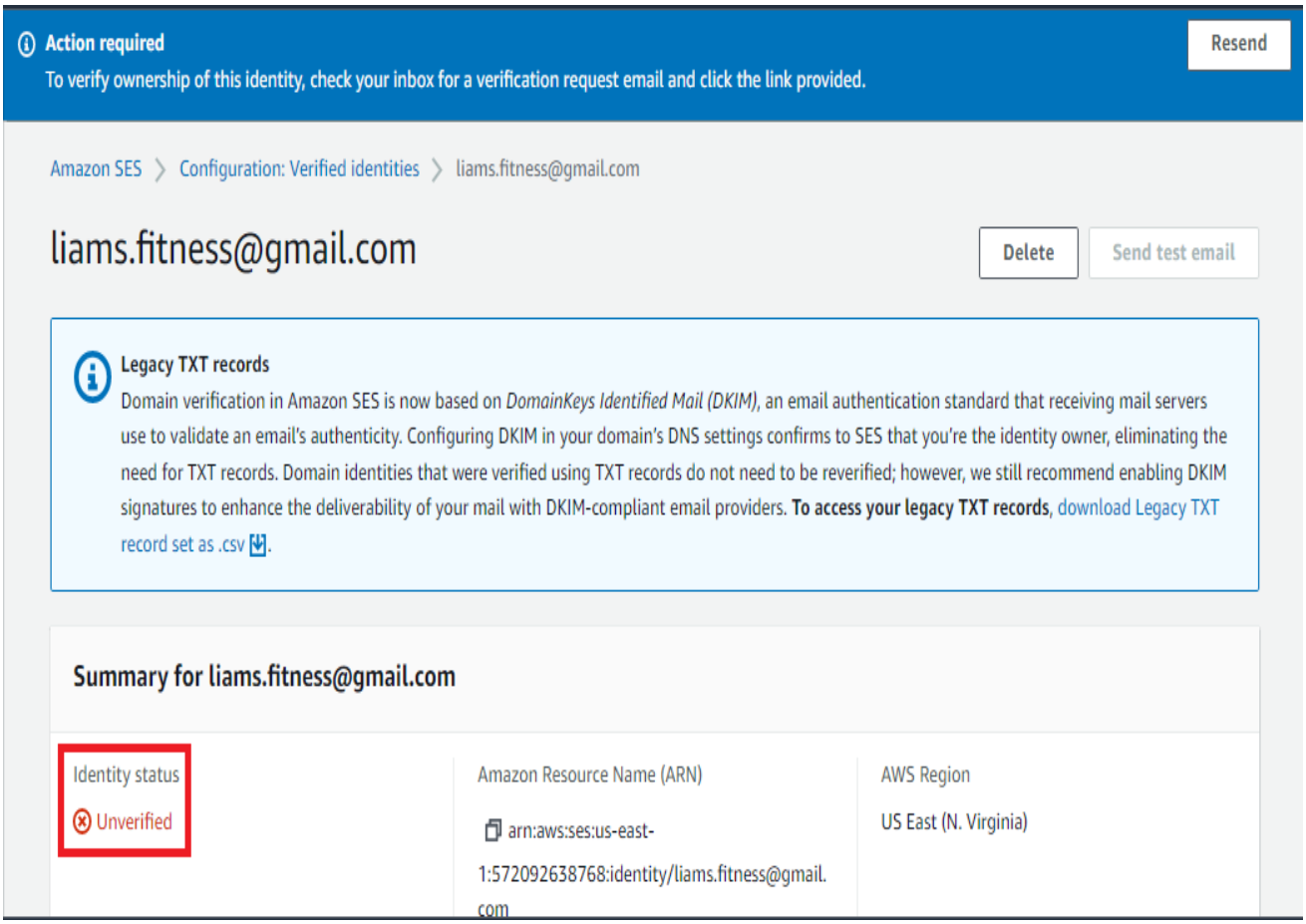
On the **Create Identity** screen, select **Email address**, then enter your email address and leave **Assign a default configuration set** unchecked.



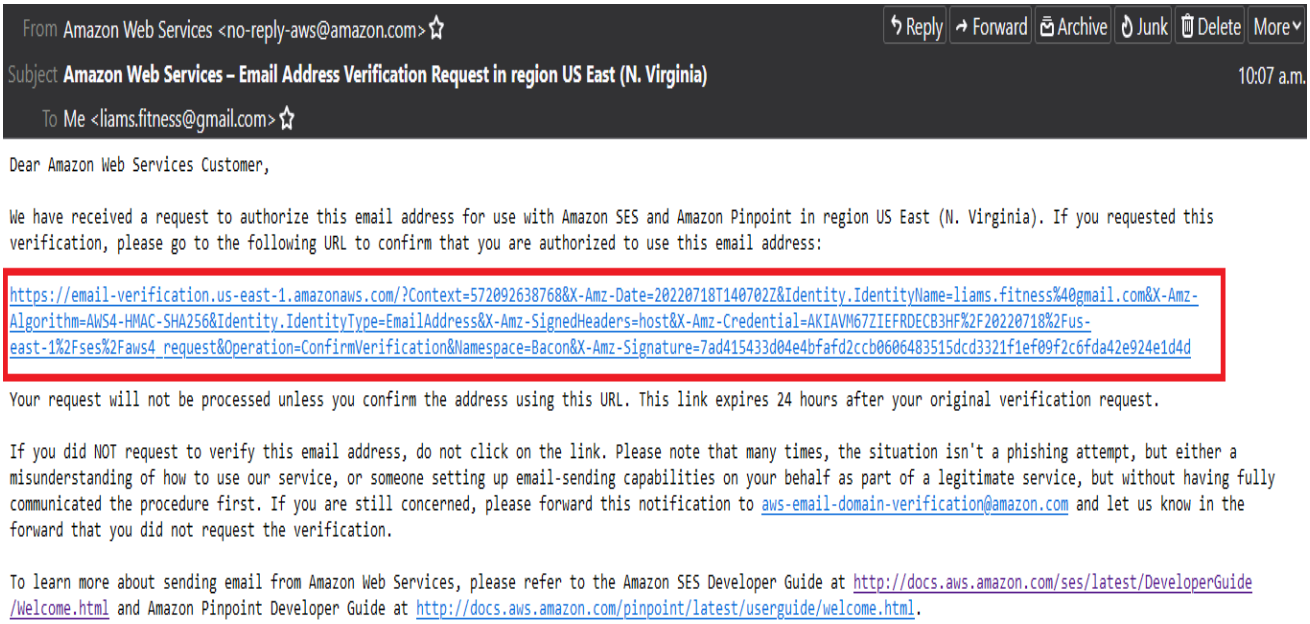
You can add a Tag if you wish, I have not. Finally, click **Create identity**



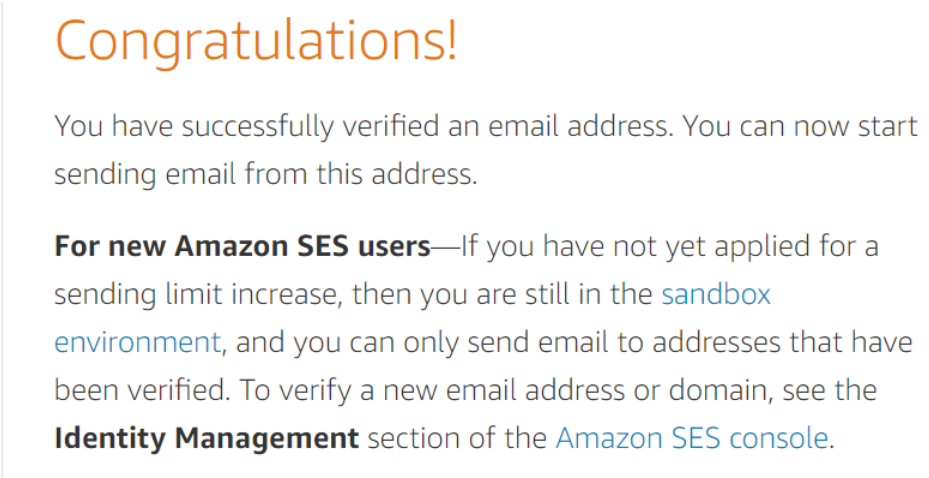
A verification email will be sent from **Amazon Web Services** which must be confirmed in order for the email identity to be verified.



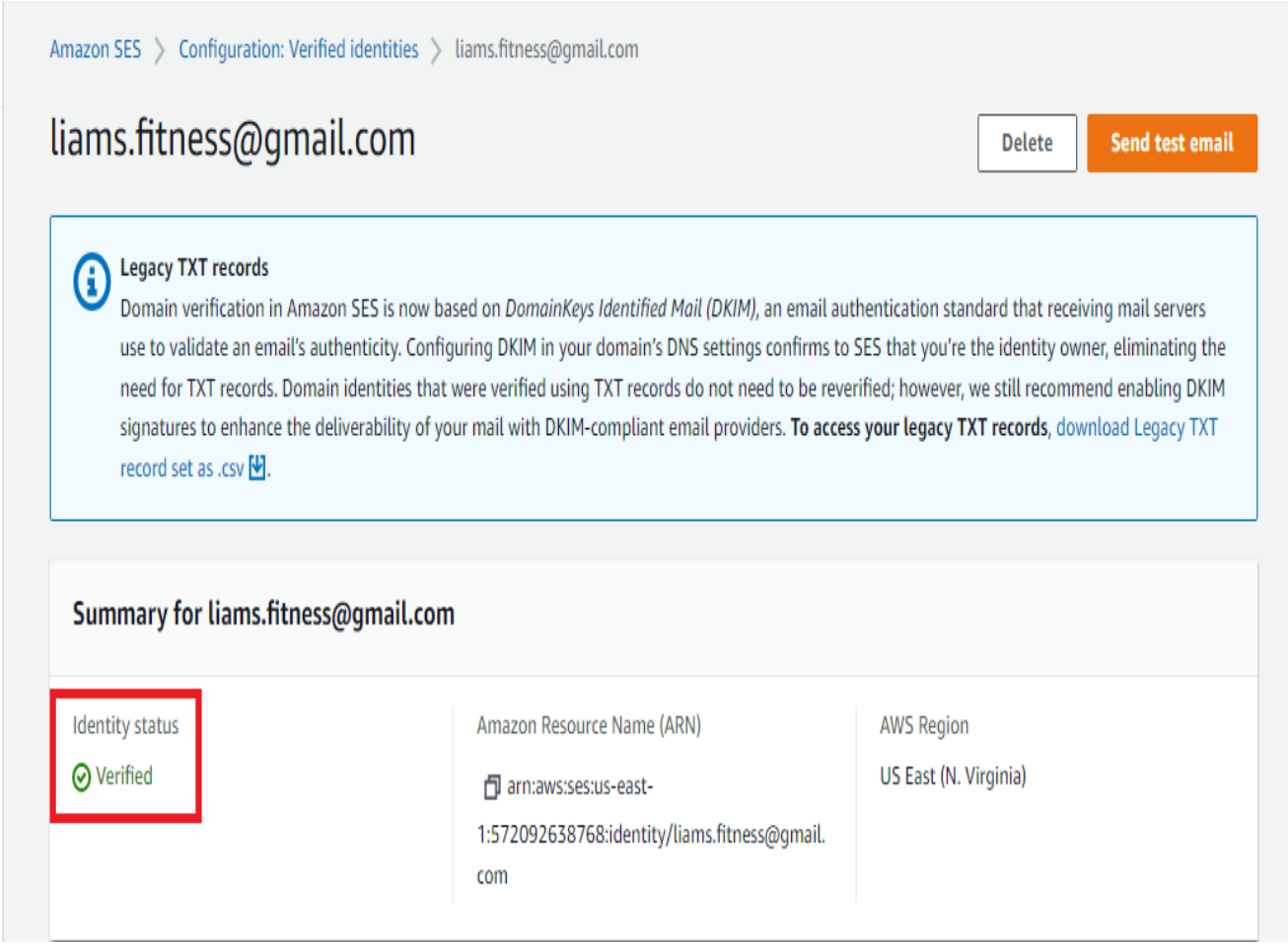
Open your email client and click the link provided to verify your email identity.



Once the link is clicked, you will be notified that the email address has been verified.



To confirm this, go back to the **Verified identities** screen, and refresh the browser window.

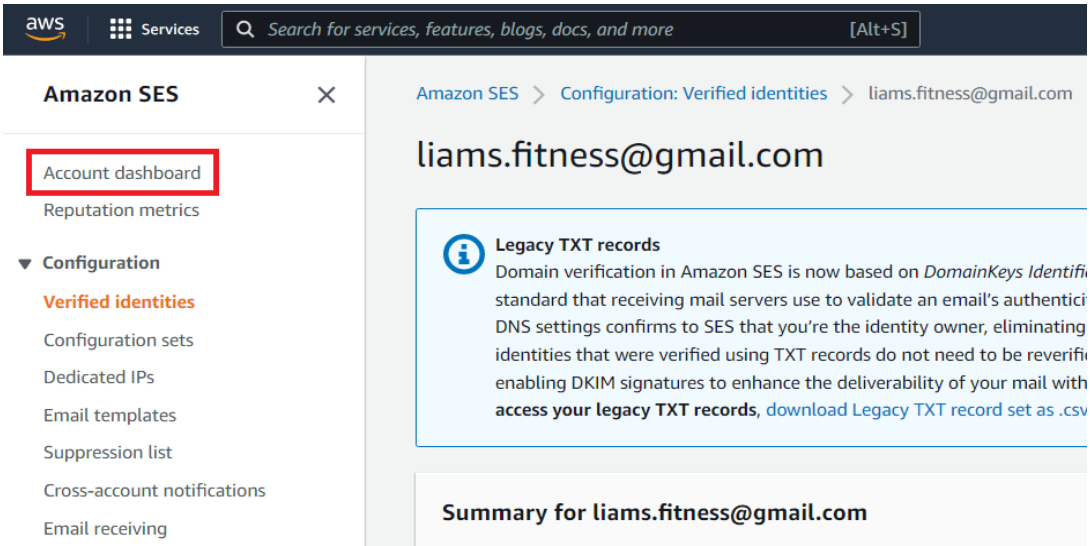


Next, we need to create SMTP credentials to access the Amazon SES SMTP interface.

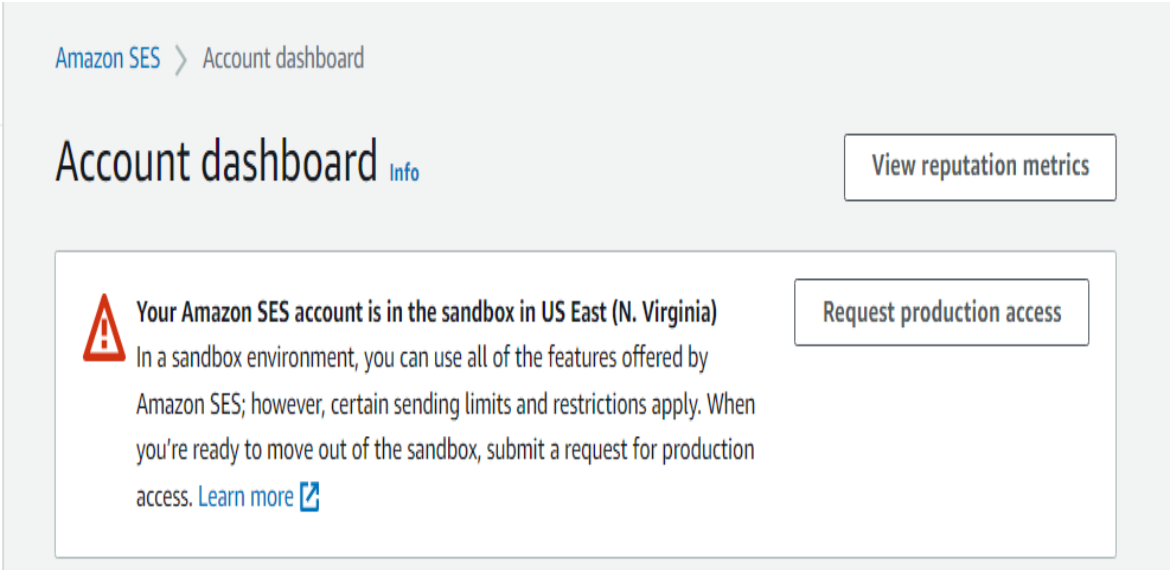
Create SMTP Credentials

To be able to access the Amazon SES SMTP interface, which is region specific, we need to create SMTP credentials. I am in the US East (N. Virginia) **us-east-1** region, so I will create SMTP credentials in that region to access that region's SES SMTP interface (endpoint). While creating SMTP credentials an IAM (Identity & Access Management) user is created with privileges to access the SMTP interface and send emails.

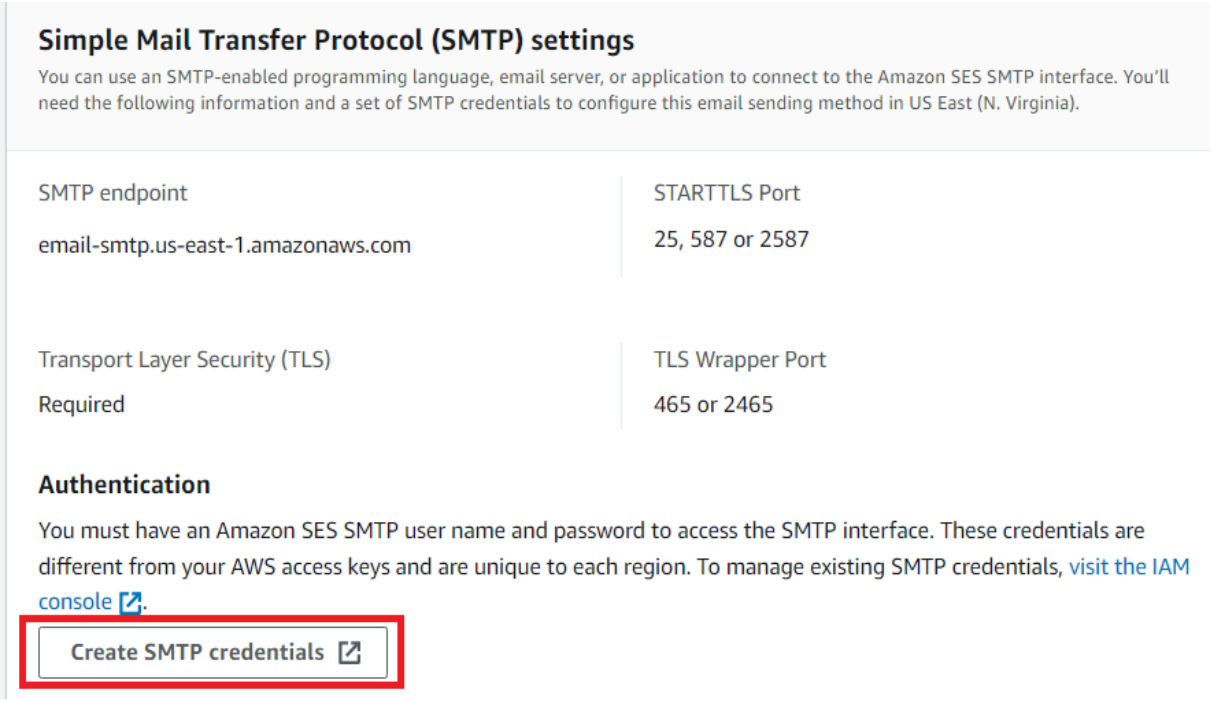
On the left-hand side of the screen, under **Amazon SES**, click **Account Dashboard**



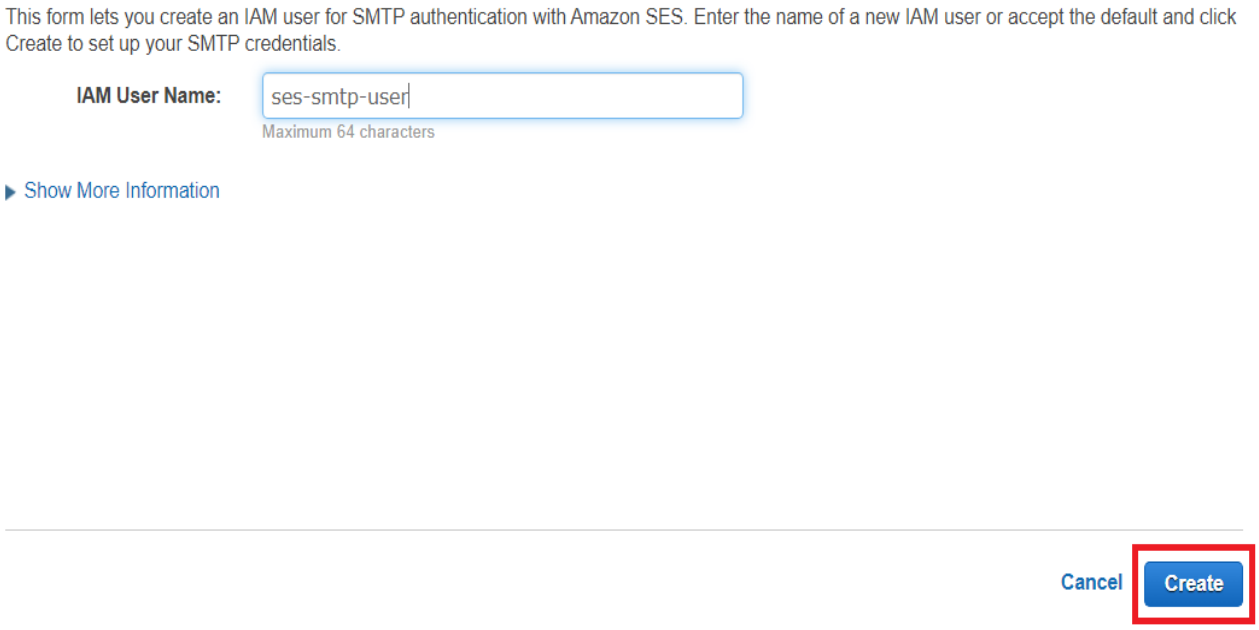
On the Account Dashboard, scroll down until **Simple Mail Transfer Protocol (SMTP) settings**



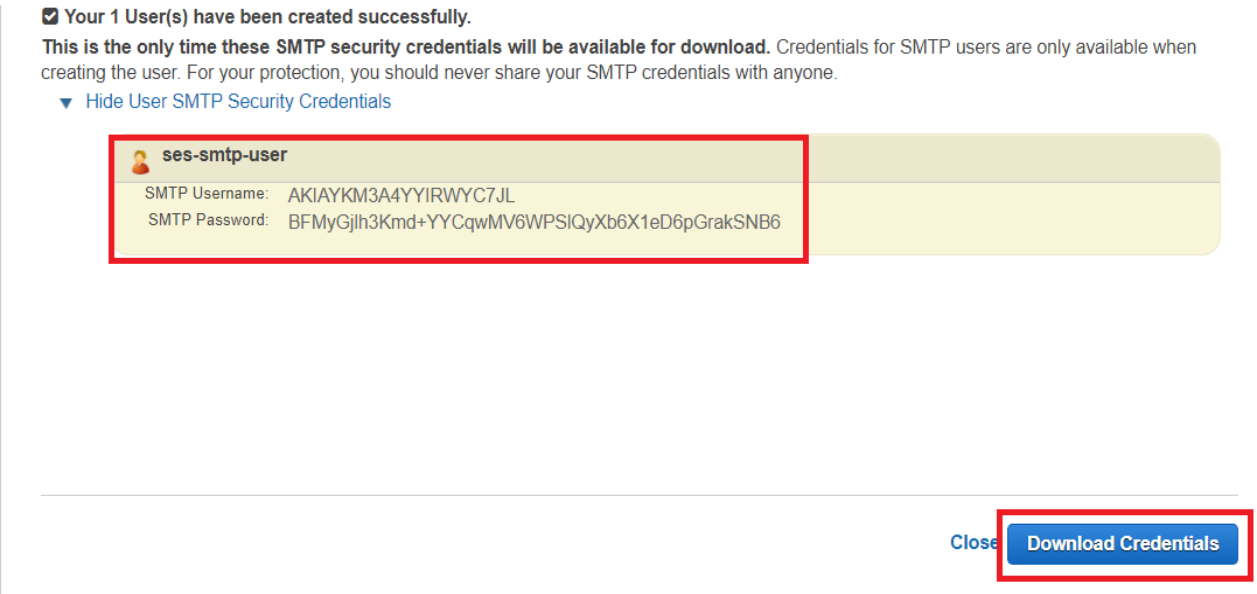
Then click **Create SMTP credentials**



On the **Create User for SMTP** screen, enter an IAM User Name and click **Create**



Now click **Download Credentials** (credentials.csv). After the file is downloaded, click **Close**.



We will need the SMTP credentials when configuring an EC2 instance to send email.

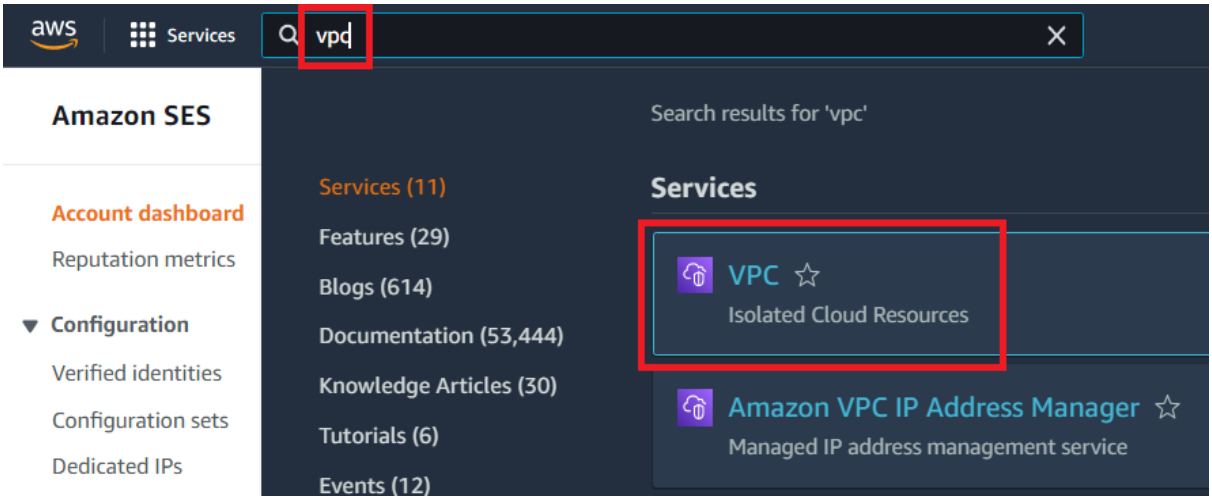
We have a verified identity and SMTP credentials. We are now ready to connect our EC2 instances to the Amazon SES service using a VPC Endpoint.

Create VPC Endpoint

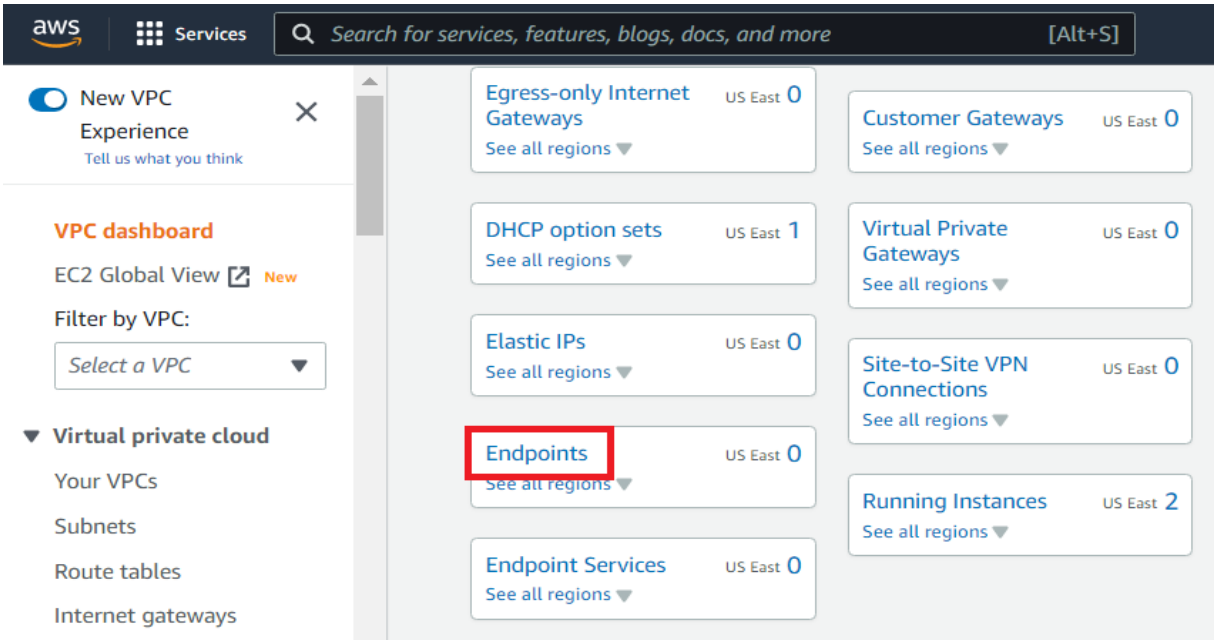
To allow our EC2 instances to send emails via Amazon SES, we need to create a VPC Endpoint which provides access to the Amazon SES SMTP interface.

VPC Endpoint creation involves a few steps. First, identity the private IP address of the EC2 instances. In my case, I am using the entire subnet so that both of my instances will have access. Second, add an inbound rule to communicate with SMTP port (587). Third, create the VPC endpoint for the Amazon SES service.

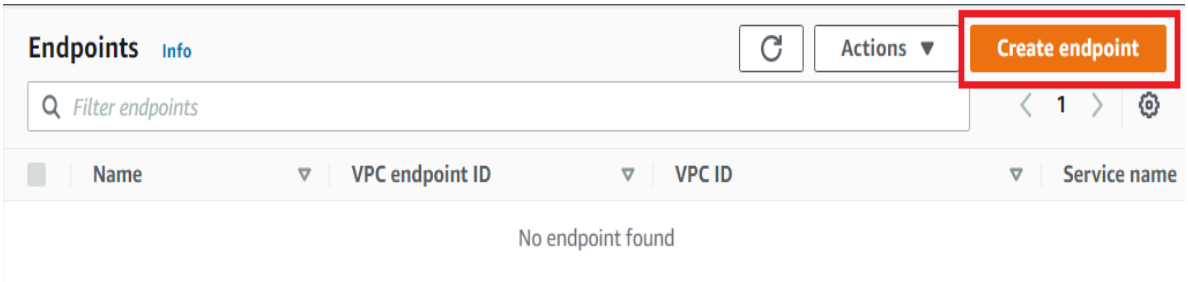
In the search bar, enter **vpc** and select **VPC** Isolated Cloud Resources.



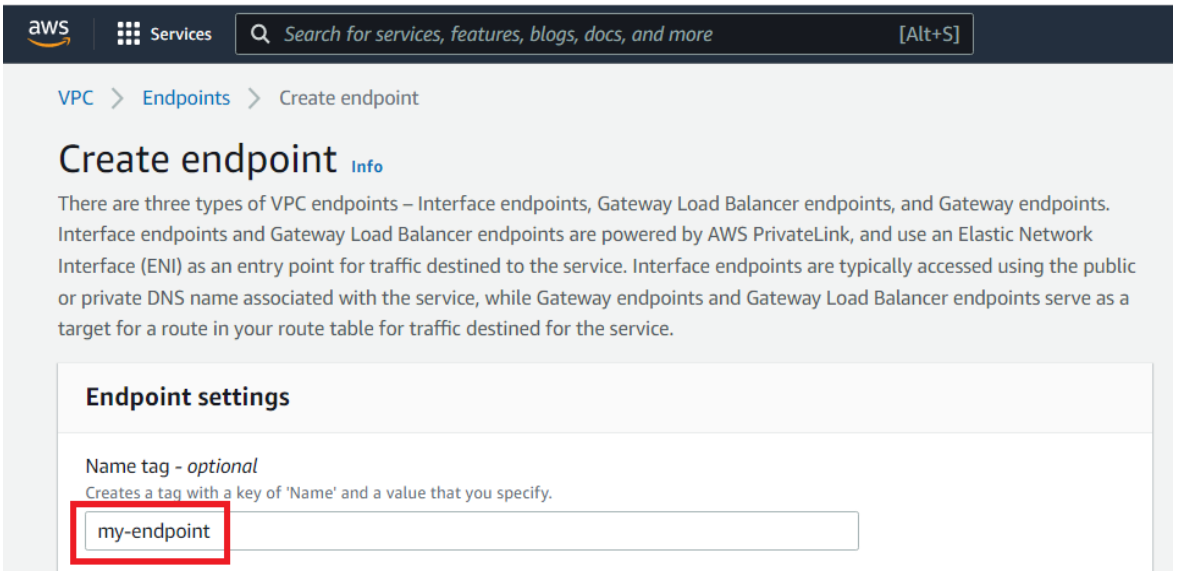
On the **VPC Dashboard**, scroll down and click **Endpoints**



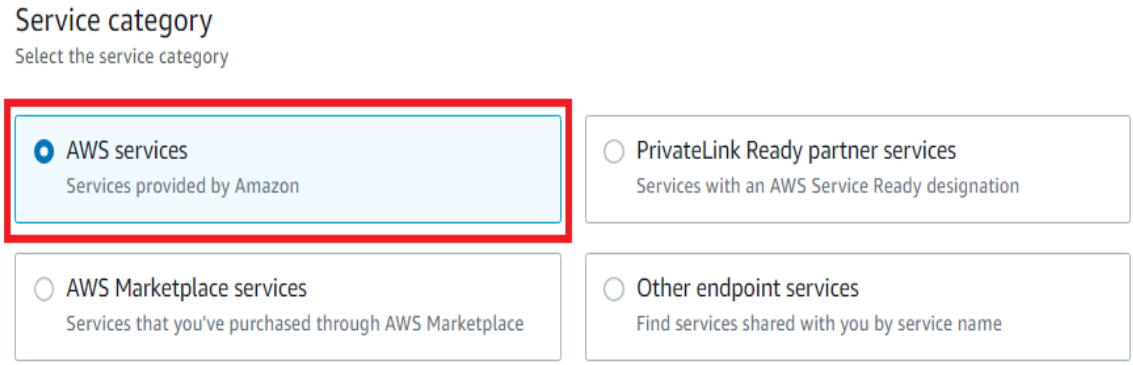
On the **Endpoints** screen, click **Create Endpoint**



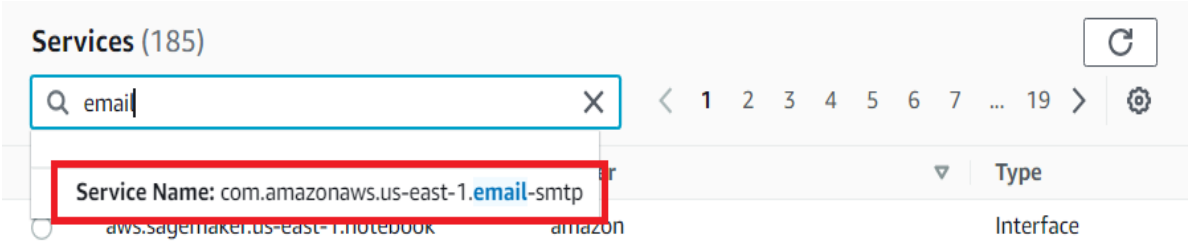
On the **Create Endpoint** screen, you can give it a name (optional)

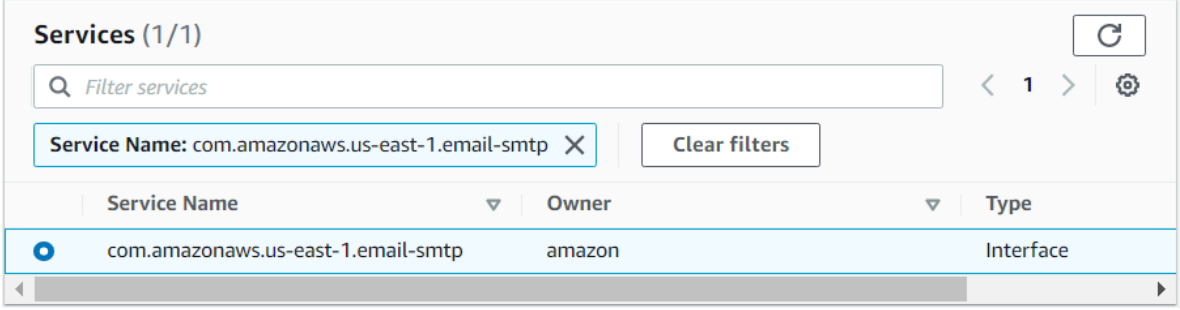


Next, scroll down to the **Service Category** section and select **AWS Services**

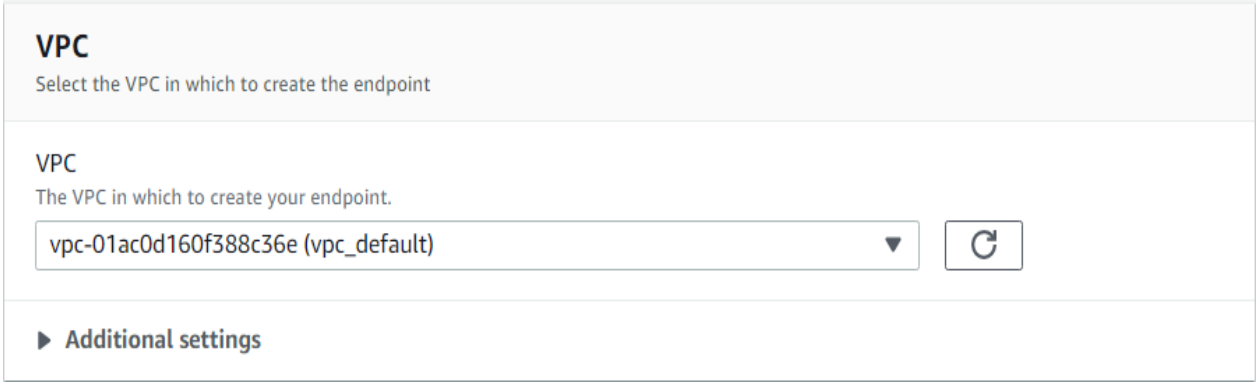


Under **Services**, enter **email** in the search box and choose the **Service Name: email-smtp**. You will notice that the service is region specific. In my case, I am in **us-east-1** (N.Virginia).

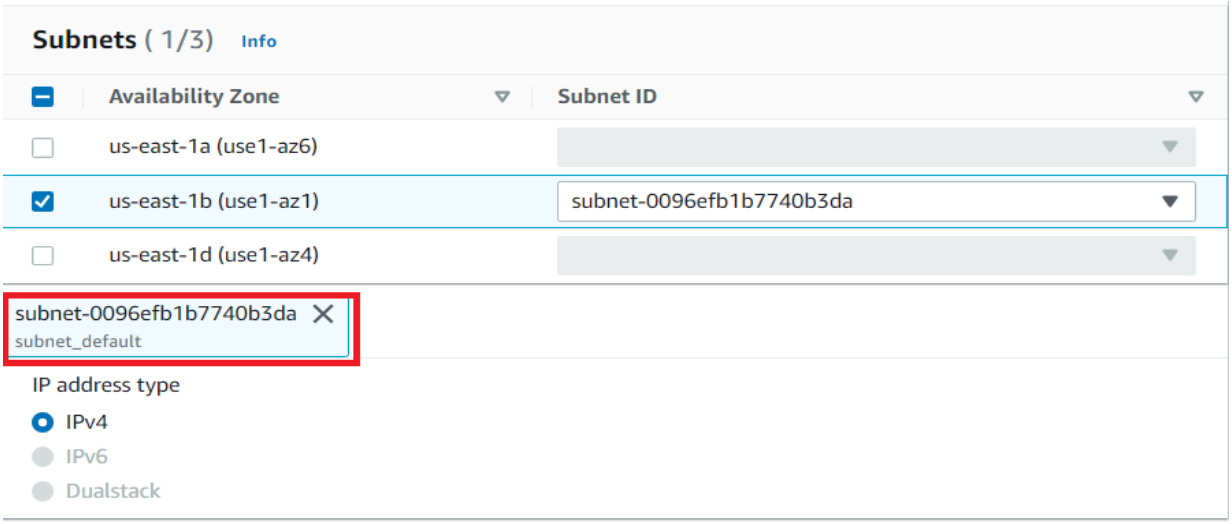




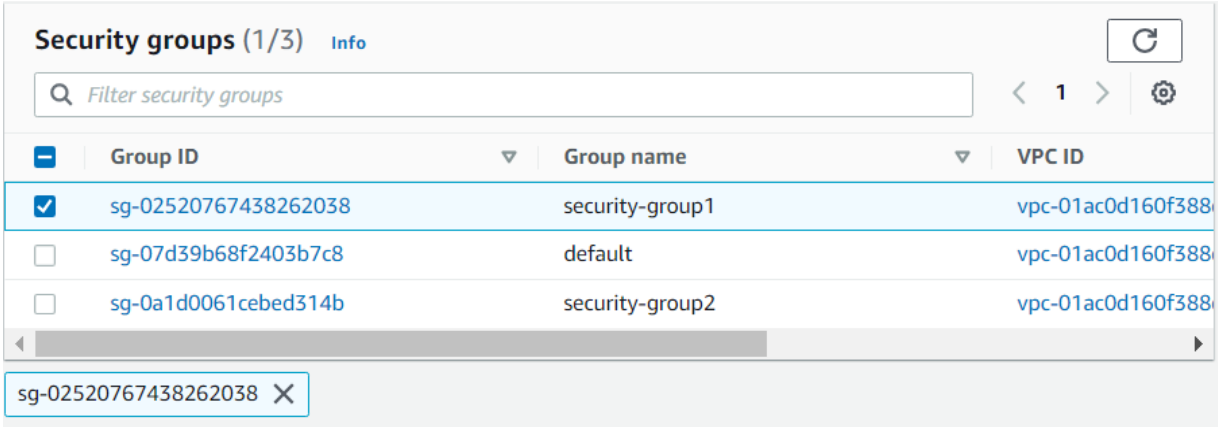
Under **VPC**, select the VPC you wish to use.



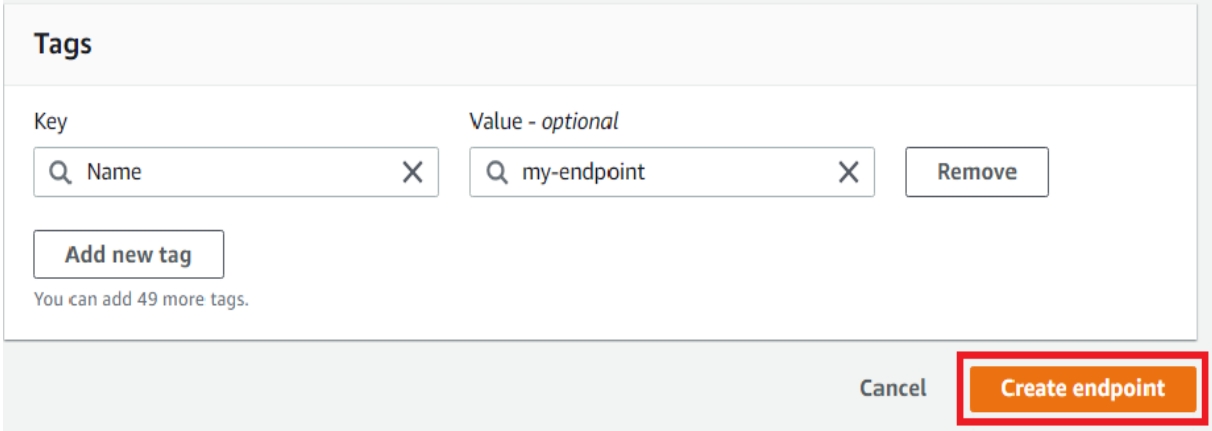
Under Subnets, choose the subnet you wish to use. For me, I will choose the subnet & availability zone (gathered earlier in the tutorial) that my EC2 instances use.



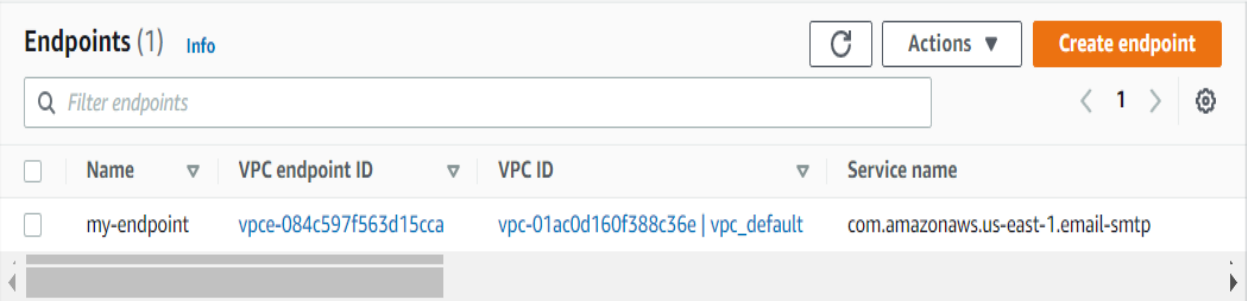
Under **Security Groups**, I have chosen **security-group1** used by both of my EC2 instances.



Finally, click **Create endpoint**



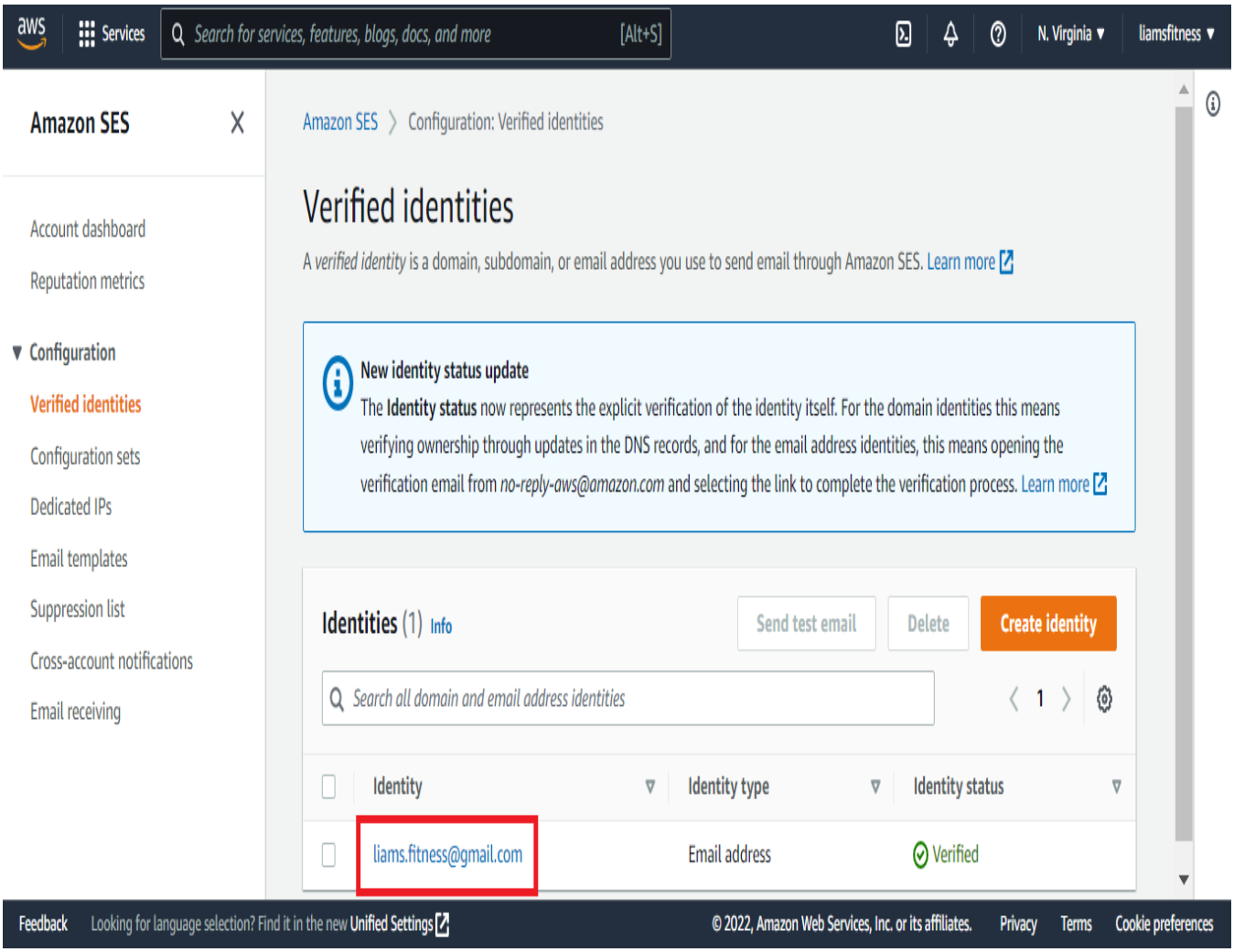
The VPC Endpoint has been created for the Amazon SES SMTP interface.



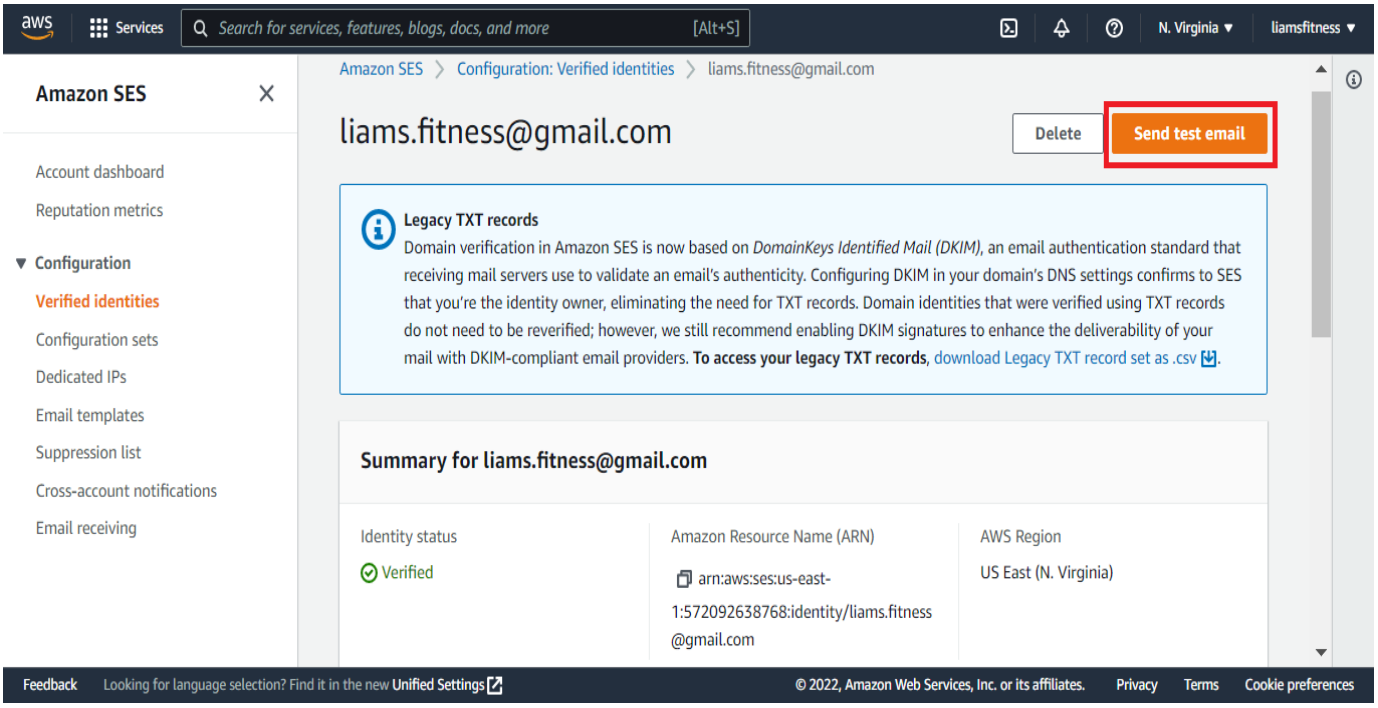
I will now return to the Amazon SES console and use the mailbox simulator to send a test email.

Simple Email Service Test

Back on the **Amazon SES Account dashboard**, ensure **Verified identities** is selected and click on the email address that is a verified identity.

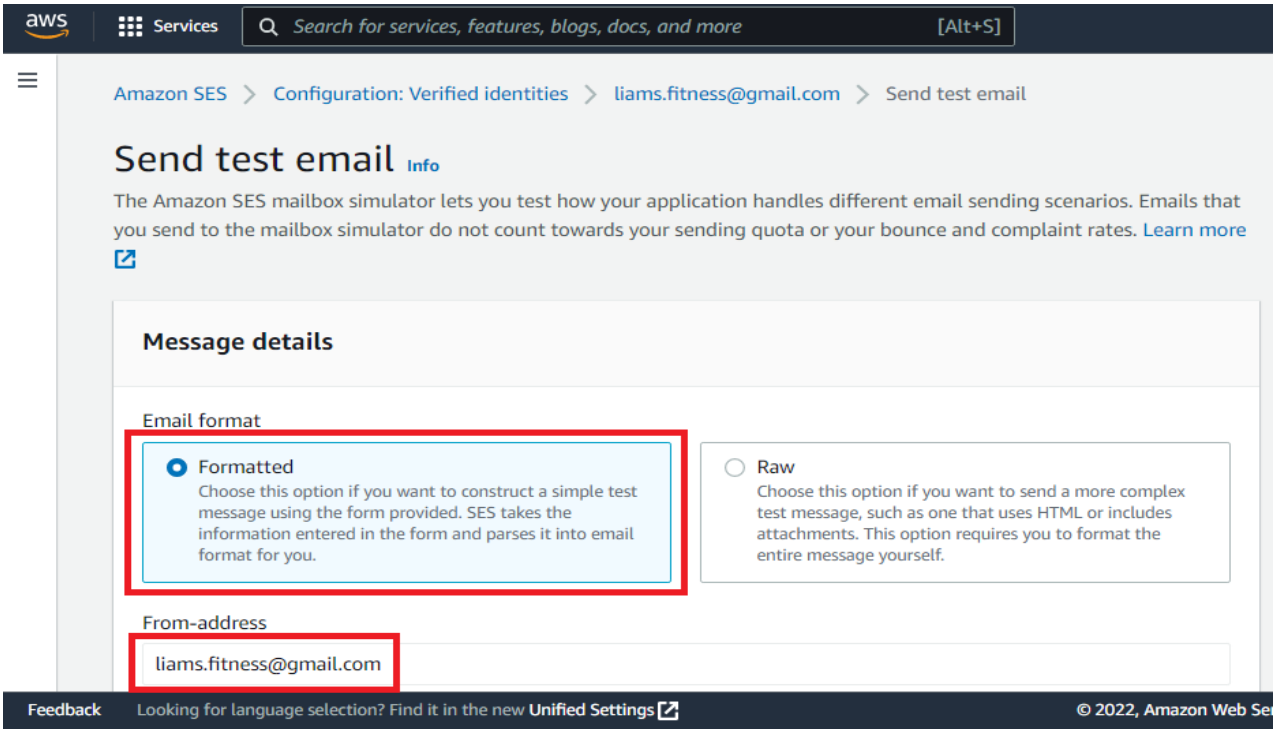


Next, I will click on **Send test email**

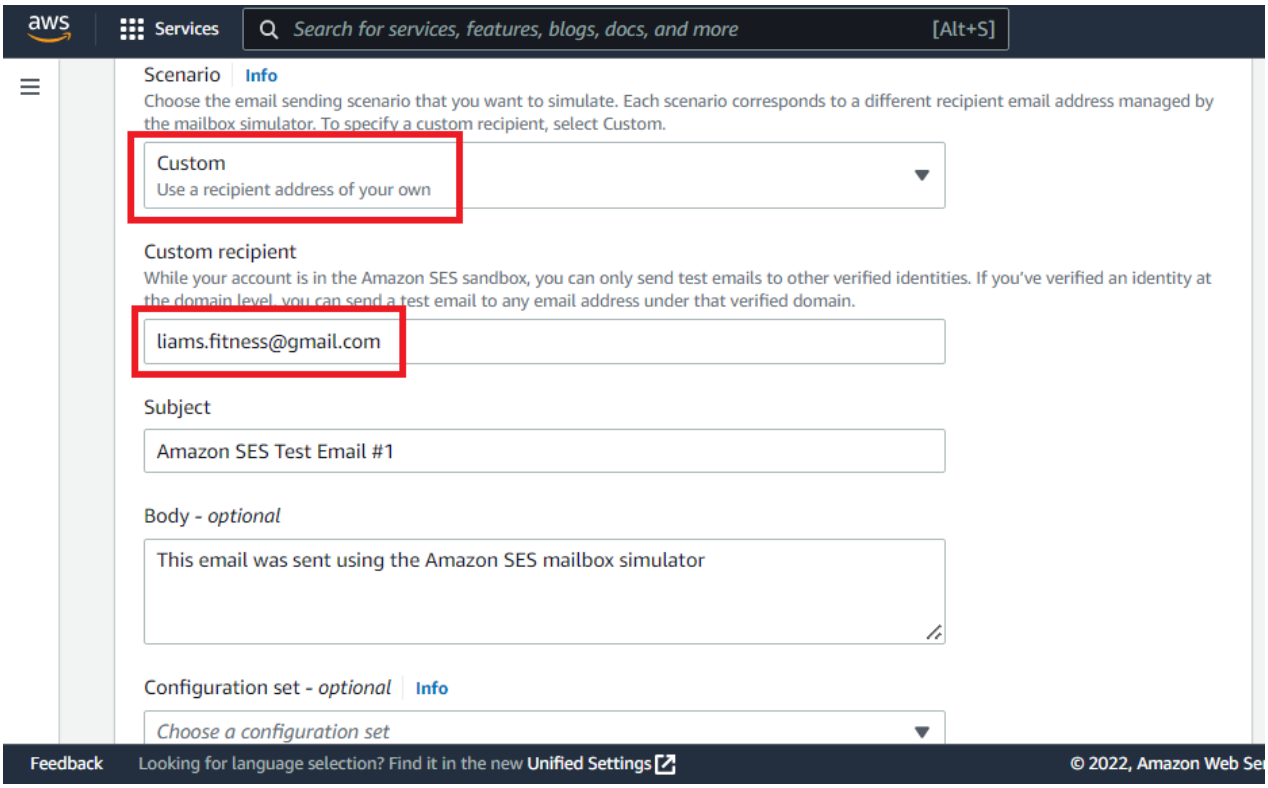


Remember that I am in the SES sandbox so I can only use my verified identity (email address) as both the sender and recipient of the test email.

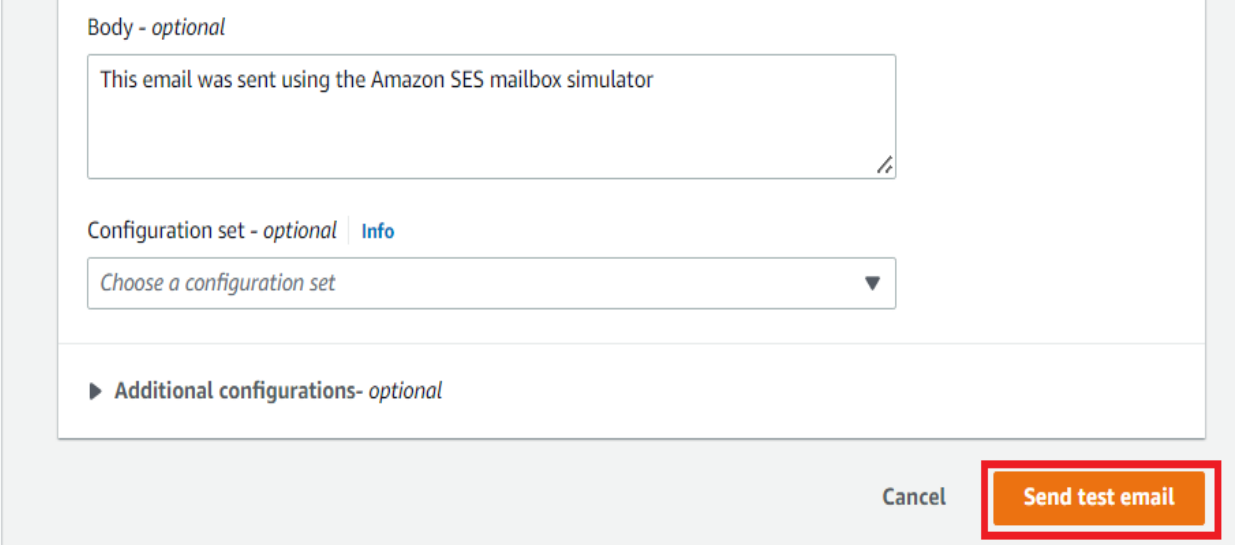
On the **Send test email** screen, ensure the **Email format** selected is **Formatted** and that the **From-address** is a verified identity.



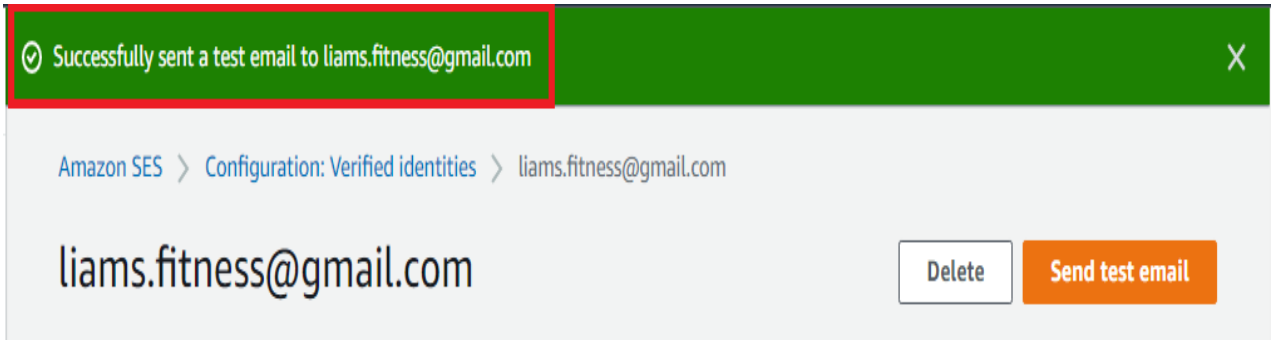
Then, scroll down the page and set **Scenario** to **Custom**, **Custom recipient** to a verified identity, a subject and, optionally, an email body. Leave **Configuration set** unselected, as it is also optional.



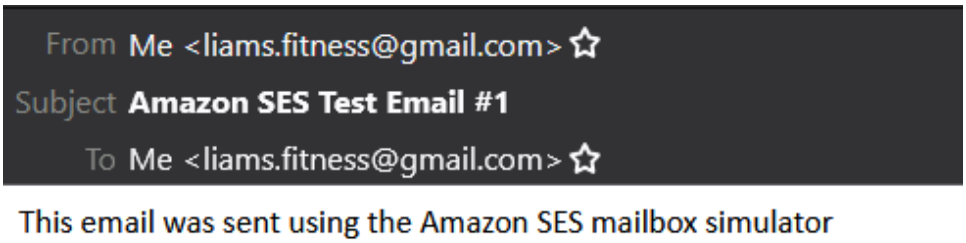
Then, scroll down to the bottom of the page and click **Send test email**



Back on the **Amazon SES console** screen, I see the successful notification about test email transmission.



I can confirm receipt by checking my email client:



I hope you've enjoyed this tutorial.

We successfully configured the components required by Amazon SES (Simple Email Service):

- A verified identity (email address)
- SMTP credentials to access the Amazon SES SMTP interface
- VPC endpoint provides access to the Amazon SES SMTP interface
- EC2 instance details for VPC endpoint:
    - security group, subnet & availability zone
- Inbound Rule for port 587 added to security group to allow email transmission

Finally, we validated the configured components by successfully sending a test email using the Amazon SES mailbox simulator.

Now that Amazon SES is ready to use, you will want to see my Postfix tutorials where I demonstrate the installation, and configuration, of Postfix as an outbound send-only email server, on an Ubuntu 20 EC2 & RHEL 8 EC2. These tutorials include how to send emails manually using **sendmail** and automatically using **cron jobs**; both via Amazon SES.

The **AWS Ubuntu 20 EC2 Postfix Install** tutorial is accessible **here**, while the **AWS RHEL 8 EC2 Postfix Install** tutorial is accessible **here**. My main tutorials page is accessible **here**.

Please be aware of the charges of $0.01 per VPC endpoint hour & $0.01 per GB for up to 1 PB monthly data processed. Although you, most likely, won't be charged for data processing, you will be charged $0.01 for every hour the VPC endpoint is active.

Back to Top